

**UNIVERSIDAD CARLOS III DE MADRID**  
**ESCUELA POLITÉCNICA SUPERIOR**  
**INGENIERÍA TELEMÁTICA**



**PROYECTO FIN DE CARRERA**

**SIMULACIÓN GRÁFICA DE SERVICIOS OSGi EN ENTORNOS  
DOMÓTICOS**

**AUTOR: JORGE HERRERA RODRÍGUEZ**

**DIRECTOR: MARIO IBÁÑEZ PÉREZ**

**Año: 2010**





---

## ***Agradecimientos***

---

Quiero agradecer este trabajo a todo el mundo que me ha apoyado y mostrado su confianza en mí, en especial a mi familia y a mis amigos. Y no puedo olvidarme de mis compañeros de carrera, que en tantas ocasiones me han ayudado.

También agradezco a Mario, mi tutor, la confianza depositada en mí para la realización de este proyecto, que me ha ayudado a aprender muchísimo, y el excelente trato recibido.

**Gracias a todos**





---

# Índice

---

1	Introducción .....	17
1.1	Planificación del Proyecto .....	20
1.2	Organización del Proyecto .....	21
2	Objetivos .....	25
3	Estado del Arte.....	27
3.1	Domótica .....	28
3.2	Lonworks .....	33
3.2.1	Introducción .....	33
3.2.2	Infraestructura Lonworks.....	35
3.2.3	Software Lonworks .....	39
3.2.4	Interfaz de Dispositivo LonMark .....	41
3.2.5	Perfiles funcionales de LonMark.....	44
3.3	X-10 .....	49
3.3.1	Introducción .....	49
3.3.2	Teoría de transmisión del protocolo X-10 .....	52



3.3.3	Dispositivos X-10 .....	59
3.4	Pasarela Residencial .....	61
3.4.1	Definición.....	61
3.4.2	Aplicaciones .....	62
3.4.3	Evolución .....	64
3.5	OSGi.....	66
3.5.1	Arquitectura OSGi.....	68
3.5.2	Framework OSGi .....	69
3.5.3	Bundles OSGi .....	71
3.5.4	Servicios OSGi.....	75
3.5.5	Implementaciones OSGi .....	76
3.6	Interfaz Gráfica y Usabilidad.....	79
3.6.1	Interfaz Gráfica .....	79
3.6.2	Usabilidad.....	80
3.7	Herramientas y tecnologías usadas .....	83
4	Estudio de Viabilidad y Presupuesto .....	87
4.1	Estudio de Viabilidad.....	87
4.1.1	La idea .....	88



---

4.1.2	Análisis del entorno .....	88
4.1.3	Conclusiones .....	89
4.2	Presupuesto .....	90
5	Análisis y Diseño del Sistema .....	93
5.1	Definición del entorno .....	93
5.1.1	Pasarela Residencial .....	94
5.1.2	Modelo de Negocio .....	96
5.2	Casos de Uso .....	96
5.2.1	Casos de Uso del Usuario.....	100
5.2.2	Casos de Uso del Administrador .....	110
5.3	Especificaciones de requisitos de software .....	112
5.3.1	Requisitos Funcionales.....	113
5.3.2	Requisitos No Funcionales .....	119
5.4	Modelo Conceptual .....	121
5.4.1	Diagrama de Clases.....	121
5.4.2	Diagramas de Secuencias .....	125
6	Implementación .....	131
6.1	Servicio DomoDevice.....	131



6.2	Decisiones de diseño .....	135
6.2.1	Dispositivos instalados en la pasarela residencial, y Dispositivos agregados a la interfaz .....	135
6.2.2	Gestión de Perfiles .....	137
6.2.3	Agregar / Eliminar Dispositivos .....	137
7	Pruebas .....	139
8	Conclusiones .....	143
9	Líneas futuras de desarrollo .....	147
10	Bibliografía .....	149
	Apéndice I: Manual de Uso .....	153
	Manual del Administrador .....	153
	Instalación de Felix .....	153
	Instalación de los bundles .....	156
	Manual de Uso para el administrador .....	159
	Manual de Usuario .....	162
	Apendice II: Descripción del CD .....	169





## Índice de Ilustraciones

Ilustración 1. Vivienda Inteligente .....	29
Ilustración 2. Capas del protocolo Lonworks .....	35
Ilustración 3. Esquema de un dispositivo Lonwork .....	36
Ilustración 4. Esquema general de los bloques funcionales dentro de un dispositivo .....	44
Ilustración 5. Diagrama de conectividad entre los perfiles funcionales de un dispositivo de luz, con sensor de nivel de luz.....	46
Ilustración 6. Diagrama de conectividad entre el Access Manager y el dispositivo Entry-Exit .....	47
Ilustración 7. Diagrama de conectividad entre los perfiles funcionales de un dispositivo de persiana, con sensor de lluvia.....	48
Ilustración 8. Ejemplo de una instalación X-10.....	52
Ilustración 9. Tiempos de sincronización de X-10.....	53
Ilustración 10. Códigos de transmisión X-10 .....	55
Ilustración 11. Estructura de un sistema X-10.....	58
Ilustración 12. Selector de un dispositivo X-10.....	59
Ilustración 13. Pasarela Residencial .....	62
Ilustración 14. Gráfico de Pasarela Residencial.....	64



Ilustración 15. Evolución de la Pasarela Residencial .....	65
Ilustración 16. Marco de aplicación de OSGi .....	67
Ilustración 17. Arquitectura OSGi .....	69
Ilustración 18. Relaciones entre bundles .....	70
Ilustración 19. Ciclo de vida de un bundle.....	74
Ilustración 20. Capas de servicios OSGi .....	76
Ilustración 21. Definición del entorno.....	94
Ilustración 22. Diagrama de Casos de Uso del Usuario .....	97
Ilustración 23. Diagrama de Casos de Uso del Administrador.....	98
Ilustración 24. Diagrama de Clases .....	122
Ilustración 25. Diagrama de Secuencias, Gestión Dispositivos Usuario.....	126
Ilustración 26. Diagrama de Secuencias, Gestión Dispositivos Administrador .....	127
Ilustración 27. Diagrama de Secuencias, Gestión Perfiles.....	128
Ilustración 28. Pantalla de bienvenido a Felix .....	155
Ilustración 29. Pantalla de instalación de Bundle.....	158
Ilustración 30. Pantalla principal .....	160
Ilustración 31. Pantalla 'Gestión Dispositivos' .....	161



Ilustración 32. Pantalla 'Gestión Dispositivos' - Administrador .....	162
Ilustración 33. Pantalla principal .....	164
Ilustración 34. Pantalla 'Gestión Dispositivos' .....	165
Ilustración 35. Pantalla 'Gestión Perfiles' .....	166





## Índice de Tablas

Tabla 1. Tabla de códigos de casa X-10.....	56
Tabla 2. Tabla de códigos de llave X-10.....	57
Tabla 3. Presupuesto por fases.....	91
Tabla 4. Caso de uso de ejemplo.....	98
Tabla 5. CU-U1, Gestión Dispositivos.....	100
Tabla 6. CU-U2, Gestión de la Iluminación.....	101
Tabla 7. CU-U3, Gestión de las Persianas.....	102
Tabla 8. CU-U4, Gestión de la Climatización .....	103
Tabla 9. CU-U5, Control de Acceso .....	104
Tabla 10. CU-U6, Control de las Fugas de Gas.....	105
Tabla 11. CU-U7, Guardar Perfil.....	106
Tabla 12. CU-U8, Crear Perfil.....	107
Tabla 13. CU-U9, Modificar Perfil .....	108
Tabla 14. CU-U10, Cargar Perfil .....	109
Tabla 15. CU-U11, Eliminar Perfil .....	110
Tabla 16. CU-A1, Autenticación.....	111



Tabla 17.CU-A2, Agregar Dispositivos.....	111
Tabla 18.CU-A3, Eliminar Dispositivos.....	112
Tabla 19.RF-01, Gestión Iluminación .....	113
Tabla 20.RF-02, Gestión Persianas .....	114
Tabla 21.RF-03, Gestión Temperatura.....	114
Tabla 22.RF-04, Gestión Fugas de Gas .....	114
Tabla 23.RF-05, Gestión Acceso .....	115
Tabla 24.RF-06, Guardar Perfil.....	115
Tabla 25.RF-07, Botón Crear Perfil .....	115
Tabla 26.RF-08, Modificar Perfil .....	116
Tabla 27.RF-09, Cargar Perfil.....	116
Tabla 28.RF-10, Eliminar Perfil .....	116
Tabla 29.RF-11, Botón Ayuda.....	117
Tabla 30.RF-12, Aviso cambio de estado de dispositivo .....	117
Tabla 31.RF-13, Autenticarse .....	117
Tabla 32.RF-14, Añadir Dispositivos.....	118
Tabla 33.RF-15, Eliminar Dispositivos .....	118
Tabla 34.RF-16, Interfaz dinámica .....	118



Tabla 35.RNF-01, Tiempo de respuesta .....	119
Tabla 36.RNF-02, Requisitos de equipos Físicos .....	119
Tabla 37.RNF-03, Comprobación del paradigma.....	119
Tabla 38.RNF-04, Entorno Java .....	120
Tabla 39.RNF-05, Presentación .....	120
Tabla 40.RNF-06, Colores .....	120
Tabla 41.RNF-07, Botones .....	121
Tabla 42.RNF-08, Documentación .....	121
Tabla 43. Pruebas .....	141







# *1 Introducción*

---

Este proyecto se basa en la realización del análisis, diseño e implementación de una interfaz gráfica para el control domótico de una vivienda.

El término Domótica proviene de la unión de las palabras *domus* (que significa casa en latín) y *tica* (de automática) [1]. Por tanto, por domótica se entiende al conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, y bienestar, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas. Estos servicios que permite

gestionar son los aparatos domésticos tradicionales como iluminación, persianas, electrodomésticos, aparatos eléctricos... [2] Se podría acabar definiendo la domótica como la integración de la tecnología en el diseño inteligente de un recinto [1], dando lugar a lo que se conoce como vivienda inteligente.

Actualmente, la domótica se encuentra en pleno desarrollo y auge. Aun así, debemos tener en cuenta que aún estamos ante los primeros pasos de evolución de este tipo de viviendas inteligentes.

En el mercado existen gran variedad de posibilidades a la hora de implantar un servicio domótico, ya que se encuentran multitud de fabricantes de dispositivos y gran variedad de protocolos de control de los propios dispositivos. En este proyecto se ha realizado un estudio detallado de los protocolos Lonworks y X-10, tanto de su funcionamiento como de sus tipos de dispositivos. Durante ese estudio hemos comprobado que estos protocolos necesitan de un estándar que proporcione interoperabilidad, de forma que dispositivos de diferentes fabricantes puedan ser integrados en una misma red de control, permitiendo así que el gran beneficiado sea el usuario final.

Para comunicarse con los dispositivos domóticos, el usuario necesita de un software específico que le permita acceder a ellos de manera sencilla a través de una interfaz gráfica. En el mundo cotidiano tenemos muchos ejemplos de interfaz de usuario, como el pomo de la puerta es el interfaz de usuario entre la puerta y la persona, o el volante, acelerador y freno son la interfaz entre el coche y el conductor. Una idea clave a la hora de diseñar

una interfaz es que ésta debe ser visible y que el usuario debe comprender intuitivamente cómo realizar la acción que desea realizar.

Por tanto, hablar de interfaces conlleva inevitablemente a hablar del usuario, ya que éste utiliza la interfaz para interaccionar con el servicio domótico implantado en su hogar. Deberemos tener en cuenta, las características del usuario, no sólo las físicas, sino también tendremos que considerar que se trata de unos sistemas nuevos a los que socialmente el usuario aún no está acostumbrado a utilizar, por lo que la interfaz deberá ser muy sencilla de manejar y de comprender. Para facilitar el uso de la interfaz por parte de un usuario, se han seguido unos criterios de usabilidad que se detallarán a lo largo de esta memoria. A la hora del diseño de la interfaz, en este proyecto vamos a tratar de llegar al mayor número de usuarios, ya que se trata de una tecnología que actualmente está en expansión, provocando que sea cada vez mayor tanto el número de viviendas domóticas como el rango de edades de los usuarios.

Para facilitar la comunicación entre la interfaz con los dispositivos domóticos, y a su vez hacer una gestión más completa de dichos dispositivos se usará la especificación OSGi (Open Services Gateway Initiative) . Es un grupo de trabajo que surgió en Marzo de 1999, cuyo principal impulsor es Sun Microsystems. Su objetivo es definir las especificaciones abiertas de software que permita diseñar plataformas compatibles que puedan proporcionar múltiples servicios. Fue pensado para aplicarse en redes domóticas.

## 1.1 *Planificación del Proyecto*

El objetivo de la planificación del proyecto es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos, costes y planificación temporal de las actividades a desarrollar. Estas actividades se dividen en:

- 1 **Evaluación de tecnologías y alternativas:** Antes de empezar a desarrollar la aplicación es necesario comprender y saber qué es la domótica y la situación actual del mercado domótico en general. Comenzaremos por los diferentes tipos de protocolos de comunicación existentes, para después indagar en los tipos de dispositivos que estos definen, con el fin de obtener las características de los mismos. Además, también deberemos comprender las tecnologías que usaremos para la implementación, como es el caso de OSGi, y buscar información sobre interfaces gráficas para comprender qué es, por qué es importante un buen diseño, y que hay que tener en cuenta para implementarla. En ese punto veremos que la usabilidad de la interfaz será un punto clave a la hora de diseñarla.
- 2 **Determinar Viabilidad:** En esta fase se realiza un estudio para saber si el proyecto que se va a realizar es atractivo o es mejor dejarlo, y se estima un presupuesto.

- 3 **Análisis y Diseño:** En esta fase se recogen los requisitos que debe cumplir la aplicación y se idean los diagramas y estructura del sistema.
- 4 **Implementación:** En esta fase se realiza la implementación real de la interfaz, atendiendo a los criterios impuestos de diseño.
- 5 **Pruebas:** En esta fase se comprueba que se cumplen todos los requisitos marcados.
- 6 **Documentación:** Esta fase tiene la misma duración que el proyecto ya que desde el estudio inicial de las tecnologías hasta el presente documento, el proceso de documentación ha sido constante.

## 1.2 *Organización del Proyecto*

- **Capítulo 1: Introducción.** En este capítulo se realiza una introducción del por qué del trabajo, su necesidad. También se detalla la planificación del proyecto.
- **Capítulo 2: Objetivos.** Se relatan los objetivos que ha ido alcanzando el proyecto hasta su culminación.
- **Capítulo 3: Estado del arte.** Detalla toda la tecnología utilizada en el proyecto, informando del estado de la misma en el mundo actual.

- **Capítulo 4: Estudio de Viabilidad y Presupuesto.** Busca contestar la pregunta sobre si resulta deseable el lanzamiento de la aplicación y nos ofrecerá una idea de a cuánto llega la cuantía económica de nuestro trabajo como ingeniero técnico de telecomunicaciones.
- **Capítulo 5: Análisis y diseño del sistema.** Es un análisis de los requisitos que debe cumplir el proyecto, tanto por parte del usuario, como de la propia interfaz.
- **Capítulo 6: Implementación.** Se relata aquellas decisiones más relevantes que se tomaron a la hora de la implementación de la aplicación.
- **Capítulo 7: Pruebas.** Se realizan pruebas que verifican la funcionalidad total de la aplicación.
- **Capítulo 8: Conclusiones.** Incluye las conclusiones que he alcanzado tras la realización del proyecto.
- **Capítulo 9: Líneas futuras de desarrollo.** Ofrece una descripción de las ampliaciones y mejoras posibles a realizar en la aplicación a corto-medio-plazo, además de nuevas e interesantes posibilidades que podrán incorporarse en un futuro más lejano.
- **Capítulo 10: Bibliografía.** Detalla las fuentes de información usadas en la elaboración del presente proyecto, numeradas por orden de aparición.



- **Apéndice I: Manual del Uso.** Ofrece una guía de uso detallada de la aplicación, orientada principalmente a nuevos usuarios.
- **Apéndice II: Descripción de CD.** Ofrece una guía del CD que contiene la aplicación, manual del programador, programas utilizados, y documentación.







## *2 Objetivos*

---

**Objetivo 1:** El objetivo principal de este proyecto es el de desarrollar un entorno de simulación gráfica para servicios OSGi que permita, al usuario que habita una vivienda, gestionar todos los dispositivos domóticos que disponga en dicha vivienda. De forma que la persona pueda interactuar con los dispositivos domóticos como apagar las bombillas, subir las persianas, aumentar la temperatura de una habitación... desde la interfaz gráfica.

**Objetivo 2:** La aplicación también tiene que permitir la ejecución en plataformas compatibles con la especificación OSGi para que éstas puedan proporcionar servicios añadidos a nuestra pasarela residencial.

**Objetivo 3:** La aplicación gráfica tiene que tener un comportamiento dinámico, es decir, que ante la instalación de nuevos servicios (dispositivos) en la casa, el sistema los reconozca “en caliente” y los pueda añadir a la interfaz para permitir su manejo por parte del usuario. Del mismo modo si un dispositivo sufre algún daño o el usuario decide eliminarlo de la vivienda, la aplicación debe eliminarlo de manera automática sin que se vea perjudicada y siga con su ejecución, sin dicho dispositivo.

**Objetivo 4:** La gestión de los dispositivos se realizará mediante una interfaz gráfica que deberá cumplir con ciertos requisitos de usabilidad que permitan crear un entorno sencillo e intuitivo adaptado a todo tipo de usuarios..

**Objetivo 5:** La aplicación debe permitir hacer una gestión de perfiles que almacenen el estado específico de un conjunto de dispositivos. Esto permitirá reconfigurar el sistema domótico con la información almacenada en dicho perfil.



## ***3 Estado del Arte***

---

Una vez definido el trabajo y explicados los objetivos, el objetivo de este capítulo es conocer el estado actual del sector en el que se desarrolla este proyecto.

En las diferentes secciones que lo componen se puede encontrar: información sobre domótica en general (elementos que la componen, beneficios y aplicaciones), información detallada sobre los dos tipos de protocolos de control más representativos (Lonworks y X-10), conoceremos su funcionamiento y los tipos de dispositivos empleados en cada uno de ellos, una introducción a la pasarela residencial de una vivienda, información detallada sobre OSGi, y por último detalles sobre interfaces gráficas, usabilidad, y las herramientas y tecnología empleadas.

### **3.1 *Domótica***

“Por domótica entendemos la incorporación, al equipamiento de nuestras viviendas y edificios, de una tecnología que permita gestionar de forma energéticamente eficiente, segura y confortable para el usuario, los distintos aparatos e instalaciones domésticas tradicionales que tiene instaladas una vivienda (la calefacción, la lavadora, la iluminación...)” [2].

En la Ilustración 1 podemos observar una sección de una casa que contiene algunos dispositivos domóticos representados en la imagen, tales como control de persianas, control de iluminación, aire acondicionado, calefacción por zonas...



**Ilustración 1. Vivienda Inteligente**

Como veremos más adelante, los elementos que componen un sistema domótico son:

- Controladores: Son los que permiten actuar sobre el sistema, bien de una forma automática por decisión tomada por centrales domóticas previamente programadas como pulsadores, teclados, pantallas táctiles o no, mandos a distancia por infrarrojos IR, por radiofrecuencia RF (hasta 50 metros), por teléfono, SMS o por PC.
- Medio de transmisión: Según la tecnología aplicada existen distintos medios, fibra óptica, bus dedicado, red eléctrica, línea telefónica, TCP/IP, o por el aire.
- Actuadores: reciben las órdenes y las transforman en señales de aviso, regulación o conmutación. Los actuadores ejercen acciones sobre los elementos a controlar en el hogar.

- Sensores: Son los que reciben datos, que pueden ser órdenes directas a los actuadores o pueden ir previamente a una central domótica, en función de la programación en ella introducida saldrá la orden final al actuador correspondiente. Ejemplos de sensores son los detectores de fuga de agua, de gas, de humo y/o fuego, de concentración de CO, de movimiento o intrusión, los termostatos.
- Elementos externos. Los elementos y/o sistemas instalados en el hogar que son controlados por el sistema domótico.

Los beneficios que aporta la domótica son múltiples, y cada día pueden surgir nuevos. Se pueden agrupar en:

- Ahorro energético gracias a una gestión inteligente de los equipos.
- Potenciación y enriquecimiento de la propia red de comunicaciones.
- Seguridad personal y patrimonial.
- Teleasistencia.
- Mayor calidad de vida (causado por todos los apartados anteriores).

Las aplicaciones de la domótica son múltiples debido a las amplias posibilidades que ofrece la domótica y a las posibles necesidades de los usuarios. Se pueden agrupar en:

- En el ámbito del ahorro energético:
  - Programación de la climatización.
  - Racionalización de cargas eléctricas: desconexión de equipos de uso no prioritario en función del consumo eléctrico en un momento dado. Reduce la potencia contratada.
  - Gestión de tarifas, derivando el funcionamiento de algunos aparatos a horas de tarifa reducida.
  
- En el ámbito del nivel de confort
  - Apagado general de todas las luces de la vivienda.
  - Automatización del apagado / encendido en cada punto de luz de la vivienda.
  - Regulación de la iluminación según el nivel de luminosidad ambiente.
  - Automatización de todos los distintos sistemas/ instalaciones / equipos dotándolos de control eficiente y de fácil manejo.
  - Integración del portero al teléfono, o del videoportero al televisor.
  
- En el ámbito de la protección personal y patrimonial
  - Detección de intrusos.
  - Detección de incendios, fugas de gas, fugas de agua...

- Alarma médica. Teleasistencia.
- Simulador de presencia.
- Cierre de persianas puntual y seguro.

➤ En el ámbito de las comunicaciones

- Control remoto.
- Trasmisión de alarmas.
- Intercomunicaciones.

Como se puede comprobar se trata de un concepto multidisciplinar que afecta prácticamente a todos los sectores de la construcción y los servicios, que pretende mejorar la calidad de vida del usuario final de esta tecnología. Se trata, por tanto de un sector a desarrollar muy importante para todos. Pero a pesar de que el término “domótica” aparece cada día en más lugares y la oferta de elementos domóticos es cada vez mayor, no debemos confundirnos, y hay que recordar que estamos en los primeros pasos para llegar a construir verdaderas viviendas inteligentes.

Uno de los campos que debe desarrollar la domótica es la interacción del usuario con el medio físico que le rodea. Este entorno serán dispositivos de medida de variables físicas (sensores de temperatura...), dispositivos de control y regulación de esas variables (actuadores de compuertas...), y dispositivos de interacción con el usuario (paneles de visualización de datos). Estos dispositivos constituyen las denominadas Redes de Control Domótico.



En la actualidad, a la hora de elegir una red de control, nos damos cuenta de la gran diversidad de soluciones posibles aportadas por diferentes fabricantes. Esto supone una gran ventaja para el usuario final ya que tendrá mayor disponibilidad de productos, pero para ello se deberá asegurar la compatibilidad entre los diferentes sistemas. Por lo que esta ventaja está condicionada al necesario establecimiento de algunos estándares.

En las dos secciones siguientes se detalla el estudio realizado sobre los dos protocolos de control más representativos, como son Lonworks y X10:

- Elegimos Lonworks porque, a pesar de que su uso sea más industrial, su arquitectura es un sistema abierto a cualquier fabricante, lo que permite reducir los costes y aumentar la flexibilidad de la aplicación.
- Elegimos X10 porque fue la primera tecnología domótica en aparecer, y por tanto la más sencilla de implementar, y por este motivo es de los protocolos más empleados en viviendas inteligentes.

## **3.2 *Lonworks***

### **3.2.1 Introducción**

Lonworks es un estándar propietario desarrollado por la empresa Echelon con el objetivo de que se pudieran implantar sistemas de control,

efectivos y de bajo coste. El estándar ha sido ratificado por la organización ANSI como oficial en Octubre de 1999 (ANSI/EIA 709.1-A-1999) [3].

“El estándar Lonwork se basa en el esquema propuesto por LON (Local Operating Network), y consiste en un conjunto de dispositivos inteligentes, o nodos, que se conectan mediante uno o más medios físicos y que se comunican utilizando un protocolo común”. Puede utilizar una gran variedad de medios de transmisión: aire, par trenzado, coaxial, fibra, o red eléctrica. Y también que se encuentra homologado por las distintas normas Europeas (EN-14908), de Estados Unidos (EIA-709-1) y Chinas (GB/Z20177-2006) así como por el estándar europeo de electrodomésticos CEDEC AIS [4].

Para el intercambio de información, Lonworks utiliza el protocolo LonTalk, el cual debe ser soportado por todos los nodos de la red. Este protocolo LonTalk posee dos características: es fiable y garantiza la seguridad:

- Fiable porque asegura la transmisión mediante una confirmación entre emisor y receptor, y dispone de un control de errores para todas las transmisiones.
- Seguro porque todas las operaciones en la red se realizan usando un sistema de autenticación de remitente.

Cada nodo de la Red de Control Domótico es autónomo y proactivo, lo que permite que puedan ser programados para enviar mensajes a otros nodos o para ejecutar acciones como respuesta a los mensajes recibidos. Por

tanto, el funcionamiento completo de la red surge de las distintas interconexiones entre cada uno de los nodos del sistema.

Alrededor de Lonworks se definieron toda una serie de componentes que lo convierten en algo más que un protocolo basado en capas (física, enlace, de red, transporte, sesión, presentación y aplicación), representadas en la Ilustración 2 [5].

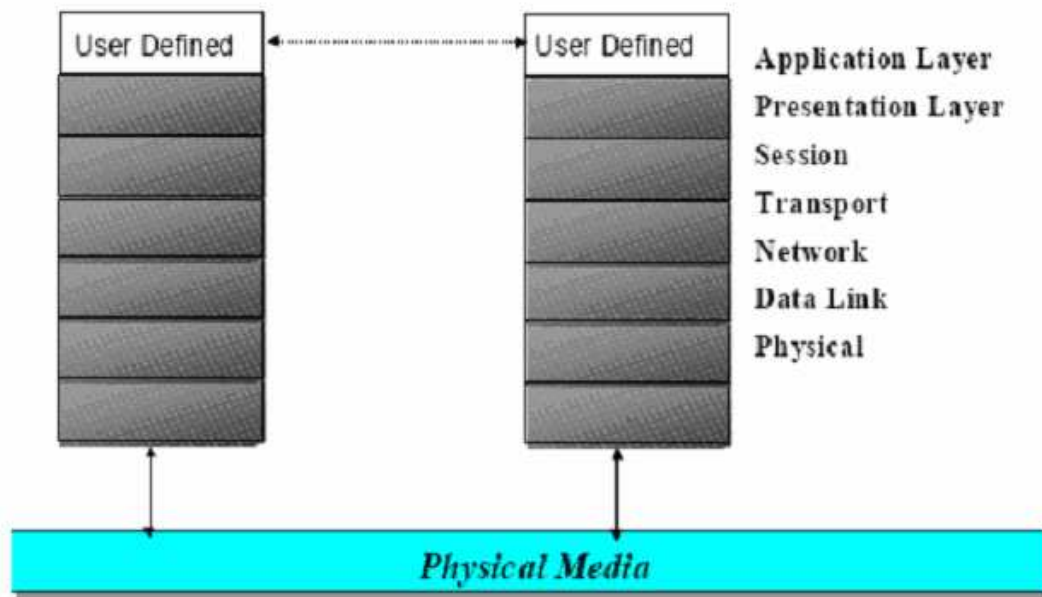


Ilustración 2. Capas del protocolo Lonworks

A continuación, analizaremos el protocolo Lonworks, y los dispositivos que ofrece.

### 3.2.2 Infraestructura Lonworks

Todos los dispositivos de la red incorporan el transceptor y el chip Neuron, aunque dependiendo de la aplicación a la que se destine, pueden incluir sensores y/o actuadores, interfaces con sensores y/o actuadores de

terceros, interfaces de comunicación serie o paralelo, etc. que les permitan interactuar con el mundo físico que deben “controlar”. En la Ilustración 3 se muestra un esquema general de un dispositivo Lonworks, observándose por un lado el Neuron chip, y por otro el transceptor (transceiver).

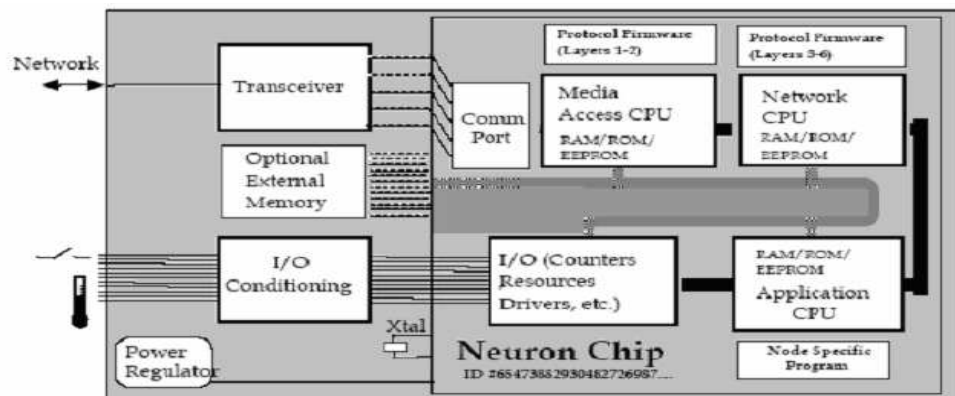


Ilustración 3. Esquema de un dispositivo Lonwork

Para ejercer ese control, los dispositivos, incluyen en el programa de aplicación una serie de algoritmos de control y proceso de datos en unos objetos de programación denominados bloques funcionales.

- ***El Chip Neuron***

El chip Neuron fue diseñado por Echelon para lograr el intercambio de información entre dispositivos conectados, cada uno con las funciones específicas para las que fue diseñado.

La ventaja principal del chip Neuron para cualquier fabricante de dispositivos, es que integra las seis capas OSI inferiores del protocolo LonTalk: desde la física hasta la de presentación, dejando al fabricante la capacidad de desarrollar cualquier aplicación utilizando la interfaz de

presentación de que dispone y despreocupándose de cualquier tarea relacionada con la implementación del sistema Lon.

Por último, añadir que todos los Neuron Chip cuentan con un identificador de 48 bits, el Neuron-ID. Éste permite identificar a un dispositivo en la red. Pero cuando el dispositivo forma parte de una red basada en dominios, el Neuron-ID deja de utilizarse como dirección de dispositivo, utilizándose, en estos casos, una dirección de red.

- ***Transceptores***

Los transceptores son, junto al chip Neuron, los dos componentes principales dentro de una red Lonworks.

Cada dispositivo de la red incorpora un transceptor, que proporciona un interfaz físico de comunicación del dispositivo con la red Lonworks. Los transceptores simplifican el desarrollo de dispositivos Lonworks interoperables. Dos dispositivos con diferente transceptor pueden comunicarse entre sí, pero será necesario interponer un componente enrutador; esto permite poder definir segmentos de redes diferenciados por el medio físico utilizado: subredes basadas en enlaces de radiofrecuencia, fibra óptica, par de hilos,...

- ***Encaminadores LON (routers)***

Los routers Lon son equipos que permiten la interconexión de dispositivos Lon instalados sobre diferentes medios (canales) a la vez que pueden ser usados para la implementación de un diseño más o menos complejo de la red, optimizando el tráfico de información por la misma.

Estos routers ofrecen otra de las ventajas del sistema Lonworks: la capacidad de soportar distintos medios de transmisión de forma totalmente transparente para el fabricante.

Hay un tipo especial de router Lon que se denomina Lonworks/IP y que permite conectar redes Lon usando una red IP como por ejemplo Internet, lo que supone una gran ventaja a la hora de disponibilidad geográfica y temporal.

- ***Direccionamiento:***

Lonworks utiliza una jerarquía de direccionamiento de 3 niveles:

- El primer nivel es el dominio. Un nodo puede ser miembro de hasta 2 dominios.
- El segundo nivel de direccionamiento es el de la subred. Puede haber hasta 255 subredes por dominio. Una subred es una agrupación lógica de nodos de uno o más canales. Los routers funcionan a nivel de subred.
- El tercer nivel de direccionamiento es el nodo. Puede haber hasta 127 nodos por subred, de manera que puede haber un máximo de  $255 \times 127 = 32385$  nodos en un dominio sencillo.

Cualquier nodo puede ser miembro de uno o dos dominios, lo que permite al nodo funcionar como gateway entre dos dominios. Los nodos también se pueden agrupar. Los grupos de nodos pueden extenderse por varias subredes a la vez dentro de un dominio.

### 3.2.3 Software Lonworks

En este apartado veremos aquellas partes que constituyen el software empleado por lonworks, como son los programas de aplicación, las herramientas de desarrollo, interfaces de red, pasarelas, servidores web, sistemas operativos de red, y las herramientas de red.

- ***Los programas de aplicación***

Los programas de aplicación que definen la funcionalidad del dispositivo se escriben en Neuron C, un lenguaje de programación basado en ANSI C con algunas características que le hacen más apropiado para aplicaciones basadas en intercambio de información a través de redes. Este intercambio de información se lleva a cabo en forma de mensajes y variables de red.

- ***Herramientas de desarrollo***

Echelon proporciona un completo entorno de desarrollo y prueba de aplicaciones en multitud de dispositivos, un gestor de red para instalar y configurar estos dispositivos y un analizador de protocolos para examinar el tráfico de la red con el fin de corregir errores y optimizar el uso de la misma.

Estas herramientas permiten a los distintos fabricantes incorporarlas en su software y ahorrar, de forma importante, en el desarrollo de herramientas propias.

- ***Interfaces de red***

Los adaptadores de red son dispositivos Lon en forma de tarjetas (PCI, PCMCIA, serial-COM, etc) que permiten a equipos PC o portátiles conectarse a redes Lon, e intercambiar información mediante el software adecuado instalado en éstos.

- ***Pasarelas***

Las pasarelas permiten interconectar redes propietarias con redes Lon, realizando las tareas de mapeo de señales y encapsulación de mensajes en las diferentes unidades de intercambio de información. Normalmente, las proporciona el fabricante del sistema externo, ya que no es un dispositivo previsto en una red Lonworks, si no más bien una forma que tienen terceros fabricantes de introducir sus equipos en una red Lonworks.

- ***Servidores Web***

Los servidores Web son un tipo especial de pasarelas que proporcionan el acceso a la información de los dispositivos de la red Lonworks a equipos que disponen de un navegador Web.

- ***Sistema Operativo de Red***

Un sistema operativo de red (NOS) proporciona todo el conjunto de servicios a las aplicaciones para la instalación, configuración, monitorización de la red Lonworks, así como una API para la programación de herramientas de gestión y mantenimiento de la red. Se trata de una plataforma que facilita la gestión de un sistema Lonworks, pero no es necesario para el funcionamiento del mismo.



El NOS que utiliza el sistema Lonworks es LNS (Network Operating System). LNS usa una arquitectura cliente-servidor y soporta clientes basados en cualquier plataforma y dispone de servidores basados en Windows.

- ***Herramientas de red***

Debido a la complejidad que pueden llegar a tener las redes de los sistemas Lonworks, es necesario disponer de herramientas de gestión, configuración y puesta en marcha de las redes Lonworks. Estas herramientas utilizan los servicios del sistema operativo LNS y entre otras podemos destacar: herramientas de integración, de diagnóstico, y de desarrollo.

### **3.2.4 Interfaz de Dispositivo LonMark**

Hasta aquí, hemos tomado una visión general de lo que es un sistema Lonworks, y antes de acabar con él vamos a introducir a la Asociación LonMark. LonMark Interoperability Association (LonMark) se forma en 1994, incluyendo a Echelon y a algunos usuarios de Lonworks. Se busca que diversos dispositivos, sin importar el fabricante, puedan ser integrados en una misma red de control interactuando unos con otros, garantizando interoperabilidad a nivel de aplicaciones. Para ello, LonMark propone una interfaz a los diferentes fabricantes de dispositivos para la implementación de las aplicaciones en los dispositivos Lon. Por tanto, debemos señalar que existen productos Lonworks, y productos Lonworks que cumplen con LonMark, los cuales llevarán el distintivo LonMark asegurando compatibilidad (como se verá en el apartado, 3.2.5).

Los elementos que componen el interfaz de un dispositivo LonMark, son [5]:

- ***Neuron ID:***

Ya vimos anteriormente, cuando hablamos del chip Neuron, que el Neuron Id es el número identificativo de 48 bits que distingue unívocamente a un dispositivo de otro, y no se puede modificar.

- ***Programa Estándar Id (SPID):***

Es el número identificativo de 8 octetos del interfaz del dispositivo. Cada dispositivo tendrá un SPID diferente y no se podrá modificar. Este identificador se utiliza para asociar el dispositivo a su correspondiente definición de interfaz.

- ***Canal del Dispositivo ID:***

Es un campo de 2 octetos, y un dispositivo que cumple con LonMark debe tenerlo inicialmente a cero y no puede modificarlo. Opcionalmente, especifica el canal al que está conectado el dispositivo.

- ***Campo de Localización:***

Es un campo de 6 octetos dentro de la estructura de datos configurables por las herramientas de red. Su uso es opcional.

- ***Cadena de Información del Dispositivo:***

Se compone de una cadena de hasta 1024 caracteres, y en un dispositivo certificado LonMark debe especificar la estructura de la propia cadena de

documentación y los perfiles funcionales que se usan por cada bloque funcional.

- ***Bloques Funcionales:***

En los dispositivos, las aplicaciones están divididas en uno o más bloques funcionales, donde cada uno realiza una determinada tarea recibiendo información (configuración y datos de entrada) que se procesa, generándose una salida. Estos datos de entrada pueden proceder de la red o de elementos de medida incluidos en el propio dispositivo, o incluso de otros bloques funcionales dentro del dispositivo. De la misma manera, las salidas se pueden enviar a otro bloque funcional del dispositivo, a la red, o actuar directamente sobre algún motor, actuador, etc. conectado al dispositivo. Estas entradas y salidas se conocen como Variables de Red y Propiedades de Configuración.

Un tipo especial de bloque funcional es el llamado Objeto Nodo (Node Object), que es usado por las herramientas de red para gestionar otros bloques funcionales, recibir información genérica del dispositivo (como alarmas) y para configurar propiedades relativas al conjunto de bloques funcionales del dispositivo. Un dispositivo, por norma general contendrá el bloque funcional Node Object, y el resto de bloques funcionales para implementar cada función que soporte. En la Ilustración 4 se puede observar el esquema general de los bloques funcionales dentro de un dispositivo, separando por un lado el Node Object (si es necesario) y por otro el resto de bloques funcionales:

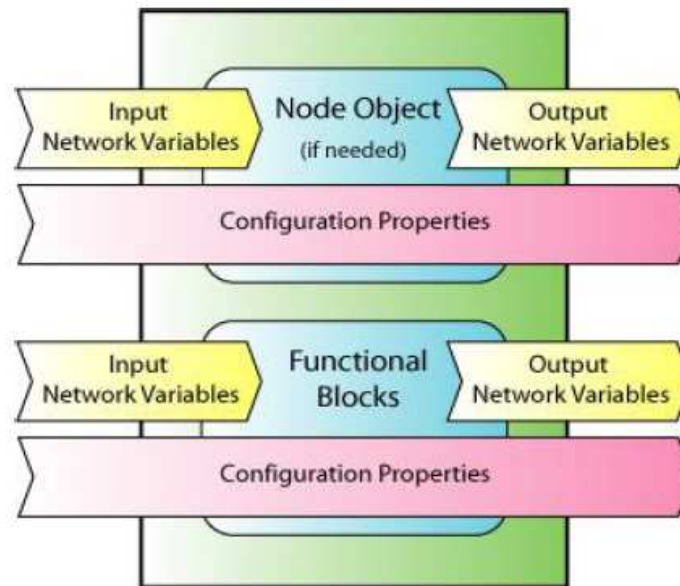


Ilustración 4. Esquema general de los bloques funcionales dentro de un dispositivo

En el caso de que el dispositivo sea muy sencillo, sólo dispone de un bloque funcional y no es necesario implementar un Objeto Nodo.

- ***Propiedades de Configuración del Dispositivo:***

Son los datos en forma de variables usados para la configuración de los dispositivos y de los bloques funcionales de la aplicación a través de la red. Las propiedades de configuración deben cumplir con unos tipos denominados Tipos Estándar de Propiedades de Configuración (SCPTs) con el objetivo de asegurar la interoperabilidad de equipos y herramientas de red.

### 3.2.5 Perfiles funcionales de LonMark

Como hemos visto, una aplicación, en cualquier dispositivo, debe implementar un bloque funcional por cada función configurada. Por medio

de los bloques funcionales, se definen unos perfiles funcionales que son los que se implementan, de forma que un perfil funcional se puede ver como una plantilla.

Como también hemos visto, las entradas y salidas por las que los bloques funcionales se comunican entre sí se llaman Variables de Red (NVs). Estas variables vienen definidas por un nombre, una dirección y un tipo. Y es gracias a estas variables de red como LonMark consigue la interoperabilidad, que mencionamos anteriormente, puesto que aquellas variables de red del mismo tipo, longitud, y direcciones opuestas pueden conectarse para permitir intercambiar información a los dispositivos Lon, aunque pertenezcan a distintos modelos de equipos o incluso a diferentes fabricantes.

LonMark propone múltiples tipos de perfiles funcionales, diferenciados por grupos industriales. En este proyecto nos centramos en: control de iluminación, acceso e intrusión, control de persianas, aire acondicionado y calefacción, control de fugas de gas. Ahora analizaremos en detalle cada uno de ellos [6]:

- ***Control de Iluminación:***

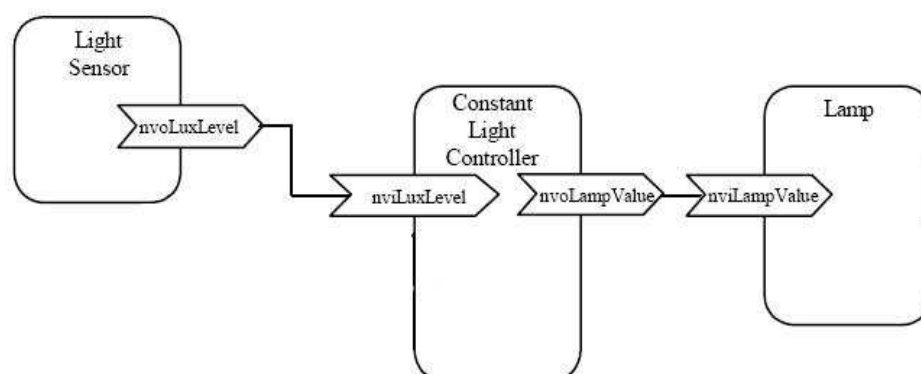
Para el manejo de los dispositivos de iluminación, LonMark tiene los siguientes perfiles funcionales:

- **Lamp Actuator:** Controla el nivel de luz emitido por el dispositivo de luz instalado. Nivel 0= off, Nivel 1=on. Este nivel de encendido varía en un rango de 8 bits, de 0 a 200, que representan el máximo

y el mínimo (0%-100%). Este valor viene marcado por la salida de un controlador, como el Constant Light Controller.

- Constant Ligth Controller: A su entrada está conectada la salida del sensor de luz (Light Sensor) que capta el nivel de luz del ambiente. Y su salida va conectada al Lamp Actuator, y será transmitida cuando sea necesario cambiar el nivel de luz del dispositivo. Además el Constant Light Controller tiene otra variable de red, que permite configurar su modo de operación (ON, OFF, DOWN-UP) [permite subir o bajar el nivel en porcentajes]).
- Light Sensor: como ya hemos visto, su salida mide el nivel de luz del ambiente, y se la manda a la entrada del Constant Light Controller.

En la Ilustración 5 podemos observar como están conectados estos 3 perfiles funcionales, y las variables de red que emplean para efectuar su comunicación:



**Ilustración 5. Diagrama de conectividad entre los perfiles funcionales de un dispositivo de luz, con sensor de nivel de luz**

Debemos destacar que éste no es el único modo de gestionar la iluminación, también se puede mediante un sensor de ocupación de una sala.

- ***Acceso e Intrusión:***

Para controlar el acceso a la vivienda, implementa:

- Entry/Exit device: maneja la apertura de una puerta, la cual puede estar en varios estados. Siempre que ocurra un cambio de estado, enviará la información al Access Manager. Además, existe un tiempo de alarma, que provoca que salte la alarma si el tiempo de intentar abrir la puerta es más largo. Y también saltará la alarma si la puerta cambia de estado y no se recibe respuesta.

En la Ilustración 6 podemos observar como se conecta este perfil funcional con el perfil 'Access Manager', y las variables de red que emplean para efectuar su comunicación:

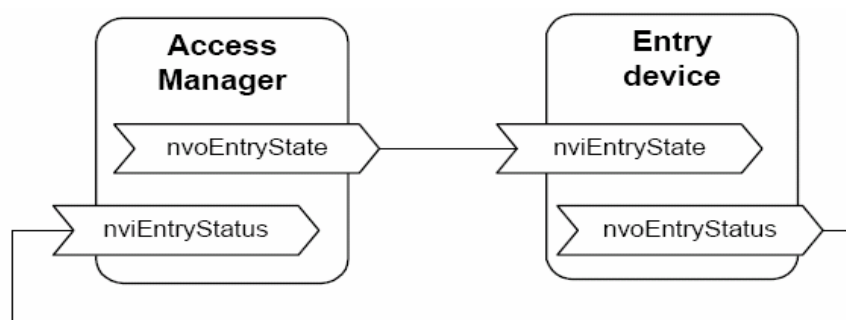


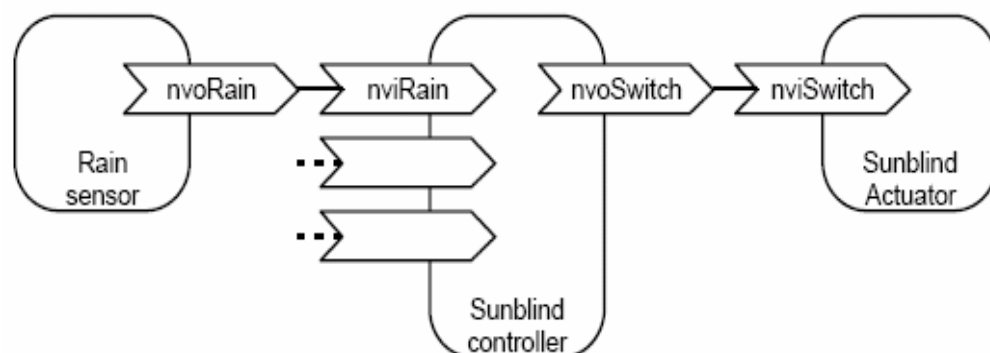
Ilustración 6. Diagrama de conectividad entre el Access Manager y el dispositivo Entry-Exit

- ***Control de Persianas:***

Como en el caso de la iluminación, también incluye un actuador, un controlador, y un sensor:

- Sunblind Actuator: A su entrada recibe los datos del Sunblind Controller, y su salida muestra el estado de la persiana. Controla motorizadamente la posición (bajar, subir) y ángulo de la persiana a la vez.
- Sunblind Controller: A él se le pueden conectar sensores para que informe al Sunblind Actuator de mover la persiana. Uno de ellos es el Rain Sensor, pero también se podría incluir el de sensor de luz para que en caso de sol subir las persianas.
- Rain Sensor: Detecta la presencia de lluvia y le informa de ello al Sunblind Controller, y después al actuador. (0%= no rain, 100%=rain)

En la Ilustración 7 podemos observar como están conectados estos 3 perfiles funcionales, y las variables de red que emplean para efectuar su comunicación



**Ilustración 7. Diagrama de conectividad entre los perfiles funcionales de un dispositivo de persiana, con sensor de lluvia**



- ***Control de Gas:***

Para el control de Gas, además de un sensor de gas, se usan las electroválvulas. El sensor de gas (Co2 Sensor) obtiene el nivel de gas detectado por el sensor, y en función de ese nivel el hardware externo al que va conectado debe actuar en consecuencia (cortar suministro de gas, cierre electroválvulas, alarma de fuga de gas...)

- ***Control del Sistema de Aire Acondicionado y Calefacción:***

Para controlar este sistema LonMark propone un controlador (Vav Controller) que conectado a un sensor externo puede modificar la temperatura del aire acondicionado o calefacción para que el usuario aclimate a su gusto la vivienda. Este controlador permita manejar temperaturas entre  $-10^{\circ}\text{C}$  y  $50^{\circ}\text{C}$  (en bits: 0 x7FFF).

## **3.3 X-10**

### **3.3.1 Introducción**

X10 fue la primera tecnología domótica en aparecer, desarrollada en 1975 por Pico Electronics of Glenrothes, Escocia, para permitir el control remoto de los dispositivos domésticos. Y en 1978 se presentó X-10 al público americano. RadioShack fue el primer cliente y aún sigue siendo uno de los minoristas más grandes de productos X-10. Desde entonces, X-10 ha desarrollado y manufacturado versiones O.E.M (Original Equipment

Manufacturer) de su Sistema de Control del Hogar para muchas compañías incluyendo Leviton Manufacturing Co., Honeywell y Norweb. Todos estos sistemas utilizan el formato de codificación X-10. Todos son compatibles y virtualmente cualquier sistema para el hogar sin cableados utiliza X-10 con módulos PLC [7].

Se trata de un protocolo de comunicaciones para el control remoto de dispositivos eléctricos. Utiliza la línea eléctrica (220V o 110V) para transmitir señales de control entre equipos de automatización del hogar en formato digital. Por lo que la utilización de X-10 no requiere de ningún tipo de cableado especial, ya para obtener comunicación se necesitan sólo dos dispositivos: un receptor, y un controlador (transmisor) [7][8].

Sus características principales son [9]:

- Ser un sistema descentralizado; configurable, no programable.
- De instalación sencilla (conectar y funcionar). Y de fácil manejo por el usuario.
- Compatibilidad con los productos de la misma gama, obviando fabricante y antigüedad.
- Flexible y ampliable.
- Emplea equipos de bajo coste.

Estas características convierten a X-10 en una de las tecnologías más extendidas para aplicaciones domóticas. Y además, su amplia gama de productos le permite ser utilizado en diversas aplicaciones:

- 
- Seguridad: intrusión, fugas de gas, inundaciones, incendio, alarma médica, simulación de presencia.
  - Confort: control centralizado / descentralizado de electrodomésticos. Manejo con mando a distancia. Posibilidad de gestión a través de ordenador personal.
  - Ahorro energético: optimización de recursos.
  - Comunicación: control telefónico remoto. Aviso de la vivienda ante incidentes.

Las principales ventajas que aporta tener instalado X-10 en una vivienda son:

- Protege el Hogar y la familia: Puede trabajar con los sistemas de alarma. Y se puede controlar y comprobar el estado de la casa a distancia.
- Añade valor a la propiedad: Una casa con un sistema domótico se cotiza más alto en el mercado inmobiliario.
- Ahorro de energía: Añadir inteligencia a la casa, además de ahorrar energía, la hace más respetuosa con el medio ambiente. Ya que al fin y al cabo se trata de un sistema que un sistema que supervisa y controla las luces y electrodomésticos apagándolos cuando no son necesarios también ahorra energía.
- Inversión protegida: Una de las grandes ventajas que tiene el sistema X10 es que es totalmente universal y por lo tanto transportable.

- Visión de futuro: De entre los varios sistemas domóticos que tratan de imponerse en la actualidad, el sistema X10 es el único que sigue vigente después de mas de 25 años y más de cien millones de aparatos funcionando por todo el mundo.

En la Ilustración 8 se muestra una instalación X-10. Como se puede apreciar la comunicación con cada dispositivo se realiza por medio de la línea eléctrica de la vivienda:

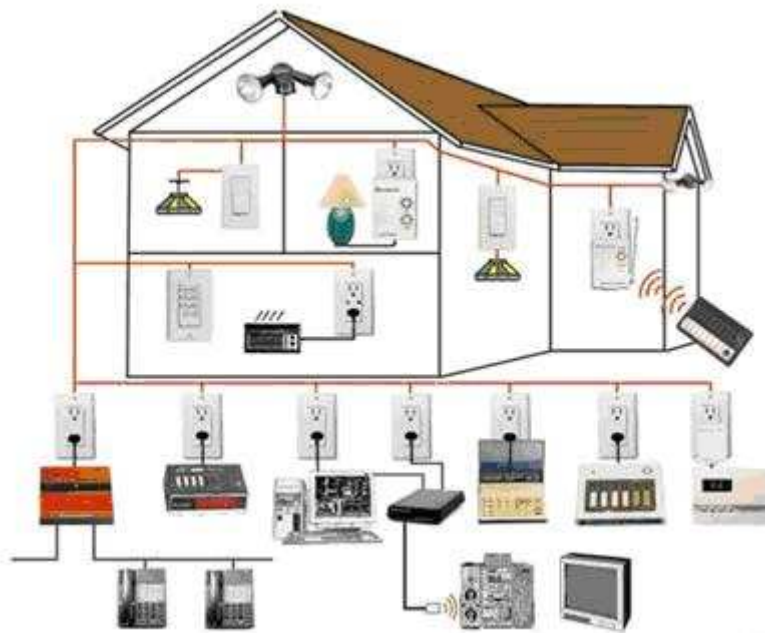


Ilustración 8. Ejemplo de una instalación X-10

### 3.3.2 Teoría de transmisión del protocolo X-10

La comunicación en el protocolo X-10 se realiza mediante el envío y recepción de señales entre transmisores y receptores, a través del cableado eléctrico de una vivienda. Tanto transmisores como receptores poseen una

interfaz que proporciona una onda cuadrada de 60 hz. y que está sincronizada al paso cero de la corriente eléctrica.

Estas transmisiones se sincronizan con el paso por el cero de la línea de corriente alterna. El objetivo es transmitir lo más cerca posible del punto de cruce por cero, ya que en ese punto el ruido de línea es menor y resulta más fácil amplificar la señal. Además se emplean filtros que atenúan las señales de frecuencia diferente a 120 Khz., ya que la línea eléctrica está expuesta a muchas interferencias, que son causadas por equipos eléctricos como TV, equipos de sonido, VCR, computadoras...

X-10 usa trenes de pulsos de 120 Khz. sincronizados con el cruce por cero de la línea. Estos trenes tienen una duración de 1ms. Cuando se quiere transmitir un '1' binario se transmite el tren de 120 Khz, como se puede observar en la Ilustración 9. Durante 1ms, y cuando se quiere transmitir un '0' no se transmite nada. Cada medio ciclo de onda de la señal de corriente alterna es capaz de transportar un bit de información.

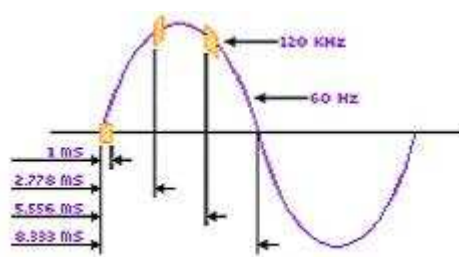


Ilustración 9. Tiempos de sincronización de X-10

La transmisión completa de un código X10 necesita once ciclos de corriente:

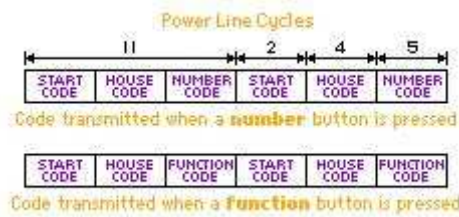
- Los dos primeros ciclos representan el Código de Inicio (1110).

- Los cuatro siguientes ciclos representan el Código de Casa (letras A-P).
- Los últimos cinco representan el Código de Llave, que será o bien el Código Numérico (1-16) o bien el Código de Función (Encender, Apagar, Aumento de Intensidad, etc...).

Este bloque completo (Código de Inicio, Código de Casa y Código de Llave) se transmite siempre dos veces, separando cada 2 códigos por tres ciclos de la corriente, excepto para funciones de regulación de intensidad, que se transmiten de forma continua (por lo menos dos veces) sin separación entre códigos. Y por último se esperan otros tres ciclos de onda antes de mandar otro bloque completo, necesarios para que el receptor mueva los datos de sus registros en cada uno de los seis pasos por cero.

Para que el receptor pueda diferenciar si se trata de un mensaje de función o de dirección, a este bloque completo de datos se le añade un sufijo.

La elección de transmitir dos veces cada bloque de información es para que el sistema ganara en fiabilidad, debido a que la línea eléctrica, como hemos visto antes, está expuesta muchas interferencias. La Ilustración 10 muestra los bloques de información transmitidos en una transmisión X-10, que como se puede observar están repetidos:



**Ilustración 10. Códigos de transmisión X-10**

El primer envío indica la dirección, y en el segundo se le indica al receptor la tarea a realizar. En X-10 estas tareas son: encender, apagar, aumento o disminución de intensidad. Una vez que el receptor ha procesado sus datos de dirección, está listo para recibir una orden de función. Al igual que se había hecho al enviar la dirección, el bloque de datos de la función debe empezar por el código de comienzo, seguido del código de la casa y el código de llave.

Cada ciclo de corriente alterna tiene un lado positivo y un lado negativo, los cuales son capaces de transportar cada uno un bit. El código de inicio (1110) se transmite dentro de dos ciclos de onda, es decir, 4 semiciclos, cada uno conteniendo un bit. Para transmitir los códigos de casa y de llave, primero se transmite un bit en el lado positivo de la señal, y en el lado negativo se transmite su bit complemento. Lo que significa que el código de inicio es el único que no se envía de forma complementaria.

La siguiente tabla muestra los códigos de casa, obteniendo las direcciones de la casa donde pueden estar instalados los dispositivos (filas) y su valor numérico (columnas):

Dirección Casa	Códigos de Casa			
	H1	H2	H4	H8
A	0	1	1	0
B	1	1	1	0
C	0	0	1	0
D	1	0	1	0
E	0	0	0	1
F	1	0	0	1
G	0	1	0	1
H	1	1	0	1
I	0	1	1	1
J	1	1	1	1
K	0	0	1	1
L	1	0	1	1
M	0	0	0	0
N	1	0	0	0
O	0	1	0	0
P	1	1	0	0

**Tabla 1. Tabla de códigos de casa X-10**

A cada unidad que exista en la casa se le asigna un código de casa y un código de llave. Esta unidad sólo responderá a la unidad central cuando el código de casa y de llave coincida con los de la petición. Si hay varias unidades con los mismos códigos, todas esas unidades responderán a esas peticiones.

A continuación, aparece una segunda tabla que muestra los códigos de llave: La primera mitad superior de la tabla representa los dispositivos identificados por números (filas) y el valor en binario de cada número (columnas), lo que representa los ‘códigos numéricos’. Y la mitad inferior representa las acciones posibles a realizar sobre los dispositivos (filas), y su valor numérico (columnas), que representa los ‘códigos de función’:



Dirección de unidad		Códigos Llave				
		D1	D2	D4	D8	D16
1		0	1	1	0	0
2		1	1	1	0	0
3		0	0	1	0	0
4		1	0	1	0	0
5		0	0	0	1	0
6		1	0	0	1	0
7		0	1	0	1	0
8		1	1	0	1	0
9		0	1	1	1	0
10		1	1	1	1	0
11		0	0	1	1	0
12		1	0	1	1	0
13		0	0	0	0	0
14		1	0	0	0	0
15		0	1	0	0	0
16		1	1	0	0	0
Códigos de función	Apaga toda unidad	0	0	0	0	1
	Enciende toda unidad	0	0	0	1	1
	ON	0	0	1	0	1
	OFF	0	0	1	1	1
	Reduce iluminación	0	1	0	0	1
	Aumenta iluminación	0	1	0	1	1
	Apaga toda luz	0	1	1	0	1
	Código extendido	0	1	1	1	1
	Petición de respuesta	1	0	0	0	1
	Notificación de respuesta	1	0	0	1	1
	Iluminación predet.	1	0	1	X	1
	Código ext. analógico	1	1	0	0	1
	Estado = ON	1	1	0	1	1
	Estado = OFF	1	1	1	0	1
	Petición de estado	1	1	1	1	1

**Tabla 2. Tabla de códigos de llave X-10**

Es importante destacar, que el protocolo X-10 estándar no dispone de ningún protocolo de acceso al medio. Sin embargo, el protocolo X-10 extendido bidireccional, que ahora empieza a surgir en los nuevos módulos, sí dispone de un protocolo de acceso al medio CSMA que hace que un emisor escuche la línea eléctrica para saber si en el momento de transmitir

ya está transmitiendo otro emisor. En este caso, esperará un tiempo prudencial para volver a intentar transmitir.

La Ilustración 11 muestra la estructura del sistema X-10. Donde podemos observar al controlador X-10 conectado a cada dispositivo (lámparas, persianas, ...) a través de los actuadores reciben las órdenes del controlador y las transforman en señales que envían a los dispositivos. También se puede apreciar que la comunicación puede realizarse a través de sensores, que reciben órdenes que envían directamente al actuador correspondiente, o previamente se las envían al controlador domótico y finalmente, éste las envía al actuador correspondiente.

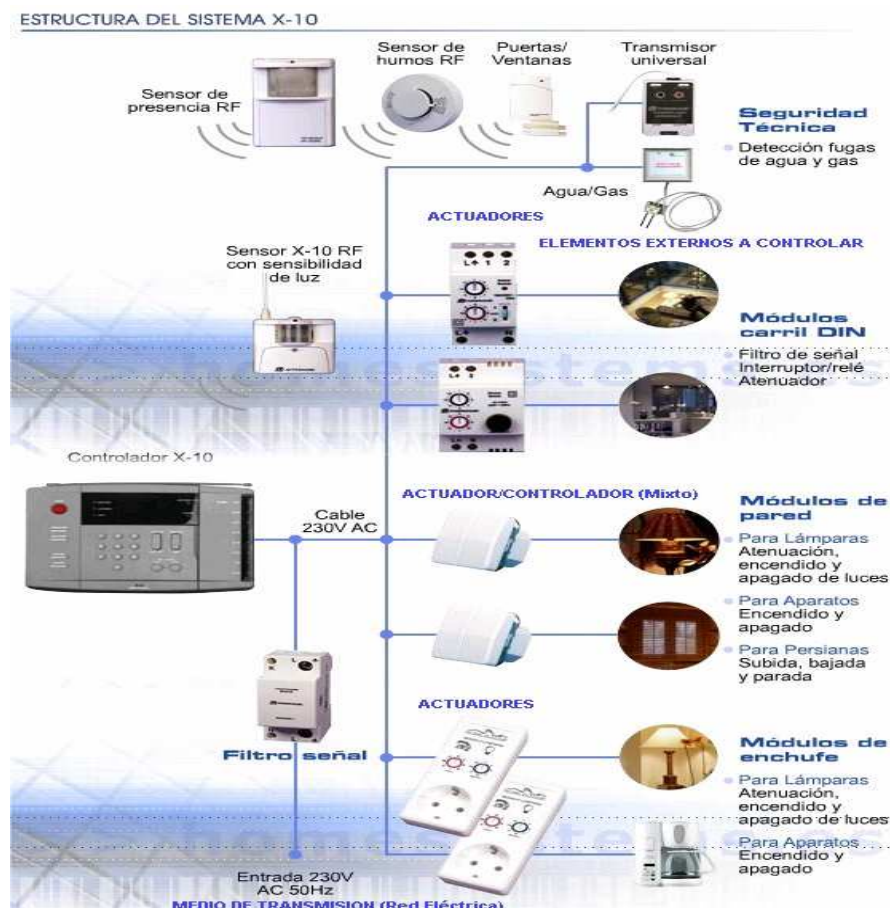


Ilustración 11. Estructura de un sistema X-10

### 3.3.3 Dispositivos X-10

Los dispositivos de X-10 son mucho más sencillos que lo que propone LonMark para sus aplicaciones Lonworks. Para ponerlos en funcionamiento basta con colocarlos en la red eléctrica y asignarles una dirección (también requiere una pequeña configuración pero que no entraremos a detallar).

Todos los dispositivos X-10 deben tener un código de casa y un código numérico. El selector donde cada usuario asigna la dirección a cada dispositivo tiene el siguiente aspecto, donde se selecciona en la parte izquierda la dirección de casa del dispositivo (código de casa), y en la parte derecha su código de llave (código numérico) [8]:



Ilustración 12. Selector de un dispositivo X-10

Cuando el dispositivo transmisor envíe una orden a través de la red eléctrica, ésta será interceptada por todos los dispositivos conectados a la red, y ejecutada por aquellos a los que va dirigida. Por ejemplo, si un dispositivo transmisor emite la orden "C9 OFF", estará indicando al dispositivo C9 que tiene que apagarse. Si hay más dispositivos con la misma dirección C9, todos se apagarán simultáneamente.

Por último, los tipos de dispositivos definidos por X-10, son [10]:

- Transmisores: Estos transmisores envían una señal especialmente codificada de bajo voltaje que es superpuesta sobre el voltaje del

cableado. Un transmisor es capaz de enviar información hasta 256 dispositivos sobre el cableado eléctrico. Múltiples transmisores pueden enviar señales al mismo módulo.

- Receptores: De la misma manera, también pueden comunicarse con 256 direcciones distintas. Cuando se usan con algunos controladores de computadoras, estos dispositivos pueden reportar su estado.
- Bidireccionales: Los dispositivos bidireccionales, tienen la capacidad de responder y confirmar la correcta realización de una orden, lo cual puede ser muy útil cuando el sistema X-10 está conectado a un programa de ordenador que muestre los estados en que se encuentra la instalación domótica de la vivienda. Este es el caso del Programador para PC.
- Inalámbricos: Una unidad que permite conectarse a través de una antena y enviar señales de radio desde una unidad inalámbrica e inyectar la señal X10 en el cableado eléctrico (como los controles remotos para abrir los portones de los garajes). Estas unidades no están habilitadas para controlar directamente a un receptor X10, debe utilizarse un módulo transceptor.

## 3.4 *Pasarela Residencial*

### 3.4.1 Definición

La Pasarela Residencial no tiene una definición estricta, de ahí que para definirla tengamos que dar las dos definiciones que se ajustan más a realidad de pasarela residencial.

Primero la podríamos definir como el dispositivo que media entre la red de la casa y la red de acceso de los proveedores. Esta definición nos da una buena idea de la ubicación de este dispositivo dentro del hogar, pero resulta demasiado genérica ya que esto incluye a muchos tipos de dispositivos. Una segunda definición trata a la pasarela como “uno o más dispositivos que conectan una o más redes de acceso a una o más redes de casa y proporciona servicios al entorno de la casa” [11]. Esta definición, que amplía las opciones de configuración es más restrictiva ya que aporta la gestión de los servicios que se proporcionan desde los proveedores, para ofrecer cierta funcionalidad para direccionamiento de dispositivos y seguridad para proteger la red interna. Los fabricantes de dispositivos han respondido a estas necesidades mejorando los productos existentes ofreciendo características como el módems para las diferentes tecnologías (RDSI, ADSL, cable, etc.), hubs multi-puerto, firewall, NAT y DHCP.

Por ello el futuro inmediato para los proveedores de servicios es la Pasarela Residencia Multiservicio, que además de ser más complejas y potentes, proporcionan varios interfaces para redes de datos y control con diferentes tecnologías. Son capaces de ejecutar diferentes aplicaciones con

requisitos de tiempo real y servicios orientados a las SOHO (u oficinas en casa) como el acceso único a Internet para varios ordenadores [12]. En la Ilustración 13 se puede observar el aspecto físico de una pasarela residencial:



**Ilustración 13. Pasarela Residencial**

### **3.4.2 Aplicaciones**

Las aplicaciones de la Pasarela Residencial son numerosas, tal y como se ha mencionado anteriormente. Quizá la que más interés presenta es la de compartir, de forma simultánea, el acceso a Internet entre varios ordenadores o equipos de entretenimiento de la vivienda como se muestra en la Ilustración 14. Las aplicaciones no están limitadas por el acceso de banda ancha a Internet sino que, gracias a la aparición de nuevos operadores y proveedores, surgirán nuevos servicios de valor añadido, e-services, más útiles que el simple acceso a Internet, destacan:

- Comunicaciones: e-mail, acceso compartido a Internet, VoIP, etc.
- Instalación *plug&play*: Debe ser sencilla y fácil de configurar, incluso la asignación de dispositivos de control domótico.
- Telecontrol y Telemetría: con aplicaciones domóticas al frente. Destacan la telegestión energética el control remoto de

electrodomésticos y equipos, el diagnóstico de los mismos y el uso de webcams que permitan observar lo que está ocurriendo en ciertas zonas o habitaciones de la vivienda.

- Seguridad: custodia y vigilancia de hogares e instalaciones, alarma de intrusión de incendios, médicas, etc.
- E-commerce: venta de productos y servicios usando la pasarela como método de acceso, por lo tanto, escaparate de los mismos, además de proporcionar autenticación de los usuarios e interfaces para métodos de pago con smartcards.
- Entretenimiento: puede servir como plataformas para vídeo/audio bajo demanda, juegos en red, charlas, char rooms, etc.

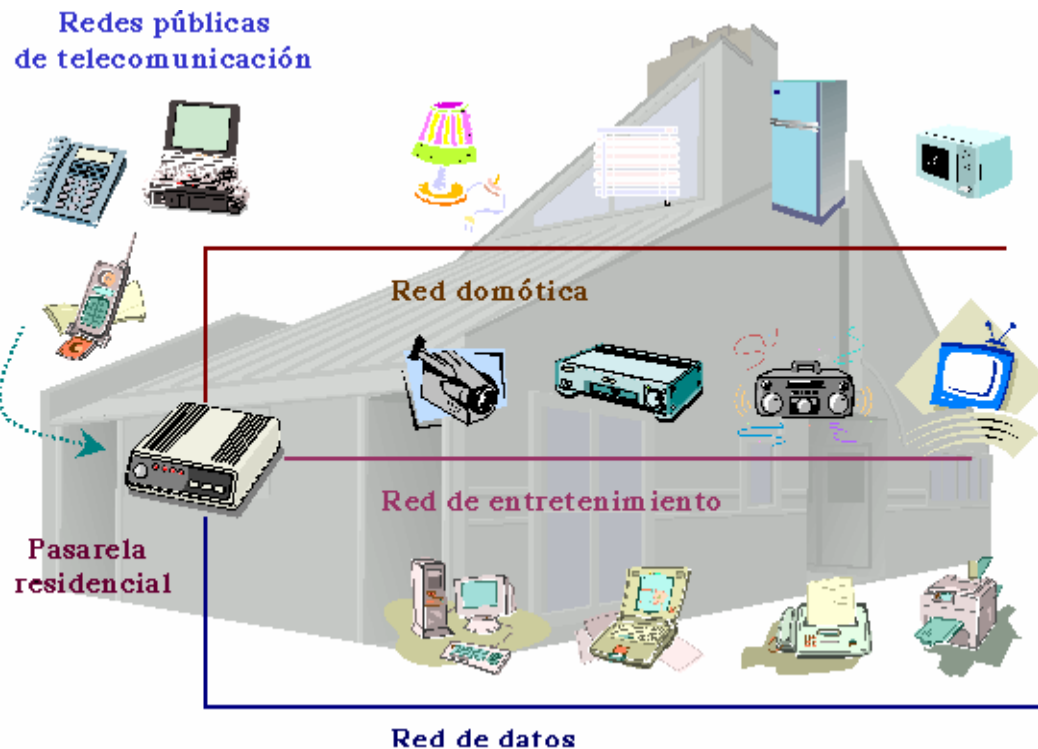


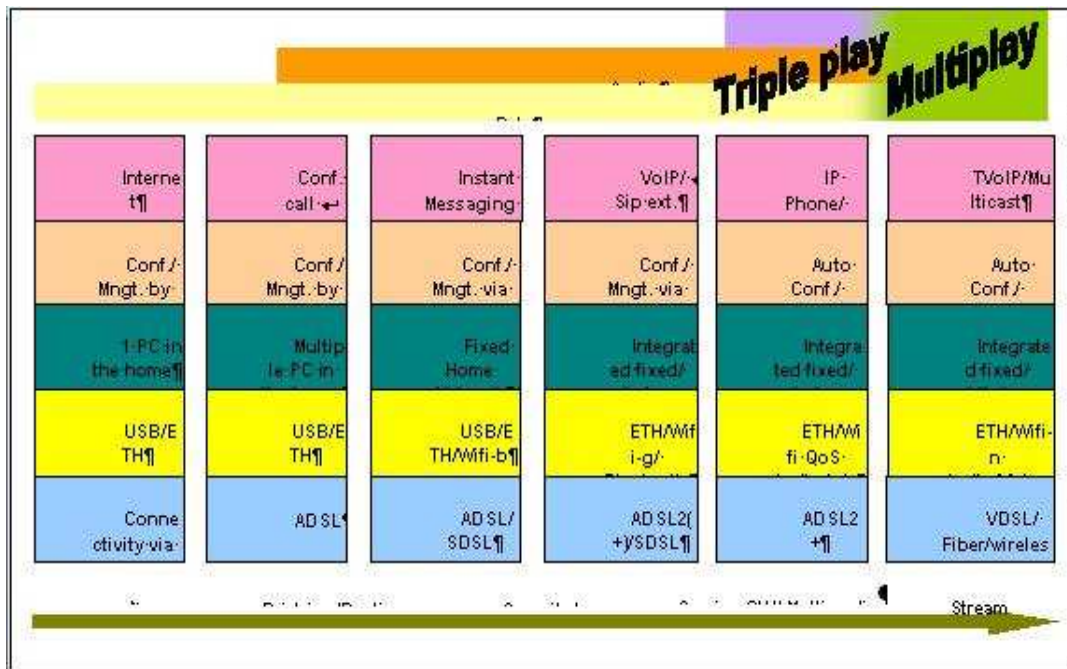
Ilustración 14. Gráfico de Pasarela Residencial

La Pasarela Residencial será programada para distribuir apropiadamente los paquetes entrantes de datos hacia cada equipos dentro de la vivienda, de igual forma empaquetará la información generada por cada unos para enviarla al proveedor de servicios correspondiente.

### 3.4.3 Evolución

Como se muestra en la Ilustración 15, este dispositivo ha sufrido una fuerte evolución en un espacio de tiempo de tan solo 8 años pasado de ser un simple modem a todo un computador capaz de gestionar múltiples servicios [13].





**Ilustración 15. Evolución de la Pasarela Residencial**

Por un lado se ha pasado de dar conectividad mediante ADSL a evoluciones de esta tecnología o algunas nuevas como la fibra óptica o las comunicaciones sin cables. También se ha producido evolución tanto en la conectividad de los aparatos como en como estos podían comunicarse con la pasarela, siendo en un principio un único PC y llegando a una red completa donde casi cualquier electrodoméstico del hogar puede estar conectado. Esta evolución se ha producido en parte por una evolución de los servicios disponibles a través de Internet que primero fueron de datos, después de voz con la aparición de herramientas de Voz sobre IP y finalmente con la llegada del video con la TV sobre IP. Este aumento de servicios ha provocado que ya no hablemos de redes de datos sino del “Triple Play” donde las redes proporcionan datos, audio y video y que estén evolucionando hacia el “Multiplay” donde no habrá diferenciación de estos tipos de datos.

Hay todavía un factor más que ha evolucionado junto con la pasarela y este es el modo de configuración y gestión. Inicialmente uno disponía de sencillas aplicaciones de usuario capaces de modificar ciertos parámetros del entonces modem o router. Al hacerse más compleja la configuración se hizo necesaria mejorar estas aplicaciones con “wizards” que ayudaban al usuario en este proceso o dispositivos que utilizan protocolos de configuración automática. El siguiente paso supone un avance tanto en la transparencia hacia el usuario como en el aumento del control por parte del proveedor sobre la pasarela ya que incluye la posibilidad de la configuración y gestión remota de dicha pasarela.

El siguiente paso en el campo de la configuración y gestión de dispositivos es de proporcionar a la pasarela de un mecanismo que proporcione, no solo una configuración y gestión remotas y automáticas sino también que permita a la pasarela adaptarse modificando su comportamiento, esto es, que la pasarela pueda proporcionar por si misma nuevos servicios según se demanden. Además la pasarela debe afrontar el reto de tener a diferentes proveedores tratando de gestionarla.

### **3.5 *OSGi***

El Open Services Gateway Initiative (OSGi) es un grupo de trabajo que surgió en Marzo de 1999, cuyo principal impulsor es Sun Microsystems.

Su objetivo es definir las especificaciones abiertas de software que permita diseñar plataformas compatibles que puedan proporcionar múltiples servicios. Fue pensado para aplicarse en redes domóticas.

Aunque OSGi define su propia arquitectura, ha sido pensada para su compatibilidad con Jini o UpnP [14].

La arquitectura de OSGi posee dos elementos fundamentales de los cuales el Service Platform está situado en la red local y conectada al proveedor de servicios a través de una pasarela en la red del operador. Este elemento será el responsable de permitir la interacción entre dispositivos o redes de dispositivos que podrían utilizar distintas tecnologías para comunicarse.

Las especificaciones OSGi definen una arquitectura SOA dentro de una máquina virtual de Java para la integración de sistemas heterogéneos.

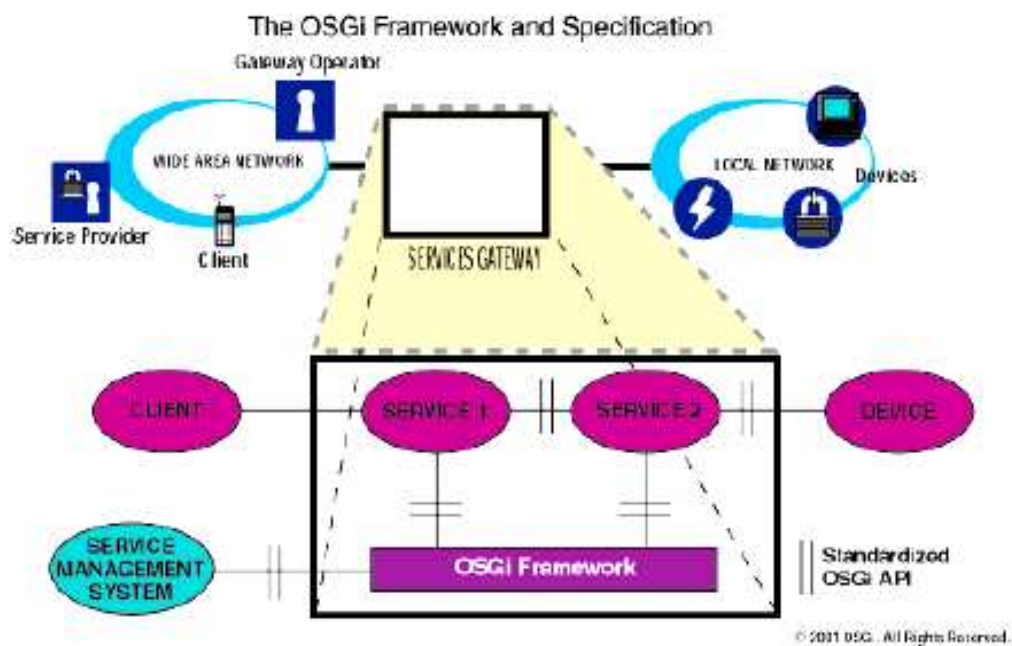


Ilustración 16. Marco de aplicación de OSGi

Las ventajas que añade la plataforma OSGi son:

- Proporciona mecanismo estandarizados para la colaboración de componentes.
- Reduce la complejidad global de construcción, mantenimiento y despliegue de aplicaciones.
- Ofrece las funciones necesarias para poder añadir y eliminar servicios dinámicamente sin necesidad de resetear el sistema
- Aparte de ofrecer los servicios comunes (Servidores HTTP, configuración, logs, seguridad, XML, ...), puede ofrecer muchos más servicios debido a su arquitectura basada en plug-ins

### **3.5.1 Arquitectura OSGi**

OSGi es un contenedor de aplicaciones orientado a servicios que proporciona funcionalidades avanzadas para la gestión del ciclo de vida de las aplicaciones desplegadas en el mismo.

La arquitectura OSGi tiene dos elementos fundamentales [15]:

- Marco de Servicio (Service Framework): Plataforma de ejecución de aplicaciones basada en Java.
- Plataforma de Servicio (Service Platform): Conjunto de servicios (bundles) que proporcionan una determinada funcionalidad a otros servicios o al usuario final, y que se ejecutan sobre el Framework. Éste, será además el encargado de permitir la interacción entre dispositivos o redes de dispositivos para su

comunicación. Los bundles están escritos en lenguaje Java, y son descargables como archivos JAR (Java Archive).

En la Ilustración 17 se pueden observar las clases e interfaces que forman parte de la arquitectura OSGi y sus relaciones:

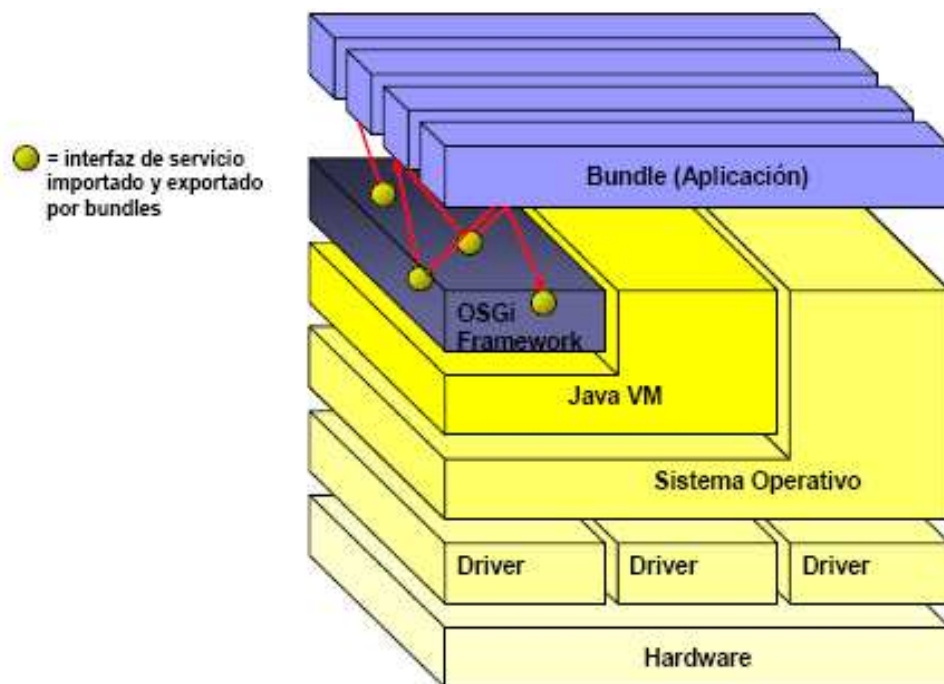


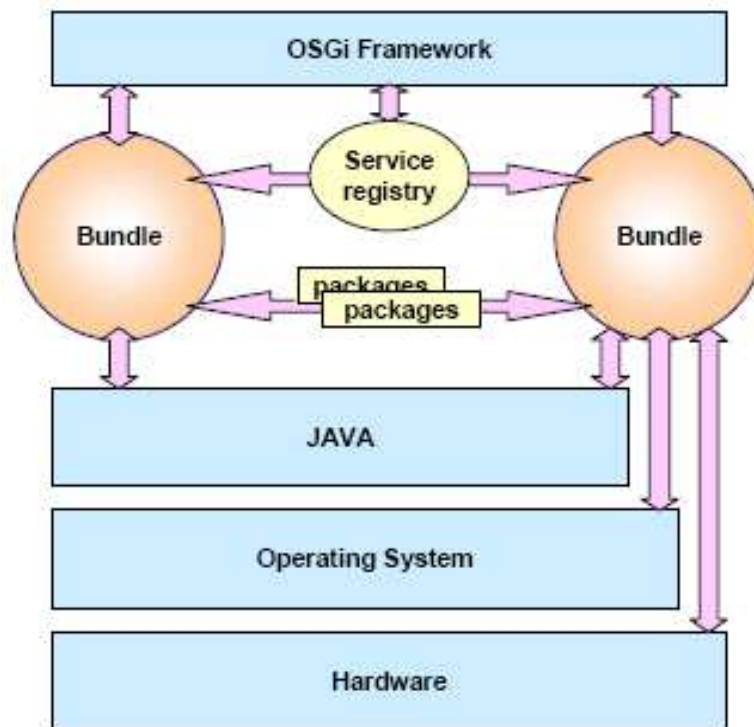
Ilustración 17. Arquitectura OSGi

Ahora veremos más en detalle este Framework, los bundles, y los servicios ofrecidos.

### 3.5.2 Framework OSGi

El framework constituye el núcleo, y define el modelo de gestión del ciclo de vida de una aplicación, un servicio de registro y unos módulos, y entornos de ejecución. Basadas en este framework han sido definidos un gran número de capas OSGi, API's y servicios.

La plataforma promociona entorno Java, de propósito general, segura y gestionada que soporta el desligue de aplicaciones de servicio extensibles y descargables e instalar bundles OSGi y eliminarlos cuando no se necesiten. Los bundles instalados proporcionan un conjunto de servicios (registrando dichos servicios) que pueden ser compartidos con otros bundles. La Ilustración 18 nos muestra las diferentes relaciones posibles existentes entre los bundles que comparten servicios:



**Ilustración 18. Relaciones entre bundles**

La plataforma consiste en un entorno de ejecución concurrente de aplicaciones (bundles), que permiten a las mismas compartir componentes (servicios) así como código, mediante exportación e importación de paquetes. Los servicios se registran en un registro de servicios para que las aplicaciones los puedan localizar. Los servicios pueden aparecer y

desaparecer en cualquier momento, sin ninguna garantía, enlazándose y desligándose en tiempo de ejecución, siendo ésta una de las características paradigmáticas de las arquitecturas orientadas a servicios.

Las aplicaciones y los componentes (que vienen en forma de Bundles para ser desplegados) pueden ser instalados, arrancados, detenidos, actualizados, o desinstalados remotamente sin necesidad de reiniciar el Framework; la gestión de los paquetes y las clases Java se especifican con un gran detalle. La gestión del ciclo de vida se hace a través de APIs, los cuales permiten políticas tanto de descargas como de gestión remota. El servicio de registro permite a los bundles detectar la incorporación de nuevos servicios o su eliminación y adaptarse en consecuencia.

El framework proporciona todo el control de dependencias entre bundles y servicios que proporcionan para facilitar la reutilización y cooperación entre los mismos. Esto hace que el entorno de ejecución sea más ligero, ya que sólo es necesario mantener una copia de las librerías en memoria y toda la plataforma se ejecuta sobre una única máquina virtual de Java, reduciendo el consumo de recursos de computación [16].

### **3.5.3 Bundles OSGi**

Un Bundle es un fichero comprimido que contiene un conjunto de clases java compiladas y otros recursos, los cuales juntos ofrecen una funcionalidad a los usuarios o facilitan servicios y/o paquetes a otros bundles. Los bundles se distribuyen como ficheros JAR. Una de esas clases java compiladas ejerce de Activator del bundle, y contiene un método start y un método stop para controlar al bundle.

En algunos casos para que un bundle pueda funcionar es necesario que se encuentren disponibles en la máquina una serie de servicios o paquetes. La dependencia a otros recursos, como los paquetes JAVA, deben estar disponibles para que el bundle pueda trabajar que las dependencias a estos paquetes se resuelven antes de arrancar el bundle.

Uno de esos recursos contenidos en el bundle es el manifiesto. Un fichero de manifiesto describe el contenido del fichero JAR y contiene información sobre el bundle. Utiliza unas cabeceras pre-establecidas para especificar los parámetros que son necesarios para instalar y activar un bundle. Las más importantes son:

- Manifest-Version: Indica la versión del manifiesto.
- Created-By: Indica al autor del bundle.
- Bundle-Name: Indica el nombre del bundle.
- Bundle-Description: Contiene una breve descripción del bundle.
- Bundle-Vendor: Distribuidor del Bundle.
- Bundle-Copyright.
- Bundle-Version: Versión del Bundle.
- Bundle-Activator: Indica cual es la clase principal del bundle la que contiene el método start.
- BundleClassPath: Directorio en el cual se buscan los ficheros jar.



- Import-Package: Indica cuáles son los paquetes o servicios necesarios para poder ejecutar el bundle.
- Export-Package: Servicios que se exportan para ser utilizados para otros bundles.

Desde que se instala un bundle, éste va variando de estado hasta que se desinstala. Todos esos estados constituyen el ciclo de vida de un bundle, y vienen definidos en la especificación de OSGi. Estos estados son:

- Installed: el bundle ha sido correctamente instalado.
- Resolved: todas las referencias que utiliza el bundle están disponibles. El bundle está listo para arrancar.
- Starting: el bundle está arrancando, ha sido invocado su método start.
- Active: el bundle está activo.
- Stopping: el bundle está detenido. Entra en este estado al invocar a su método stop.
- Uninstalled: el bundle ha sido desinstalado, y su ciclo de vida ha terminado.

La Ilustración 19 nos muestra los diferentes estados por los que va pasando un bundle durante su ciclo de vida:

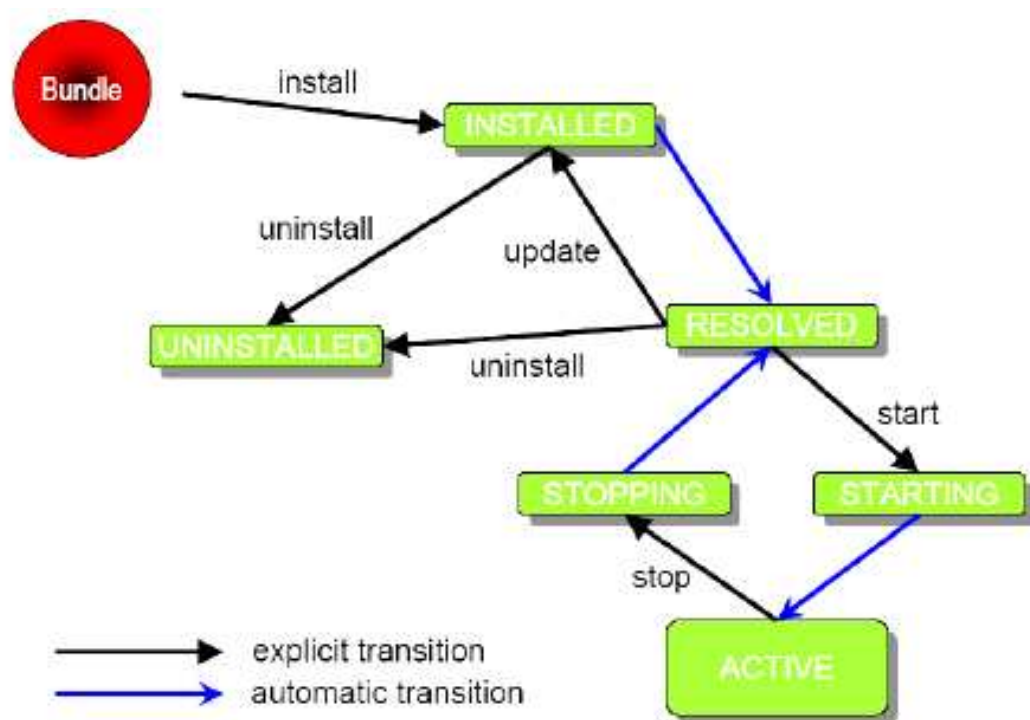


Ilustración 19. Ciclo de vida de un bundle

A la hora de definir el ciclo de vida de un bundle hay que tener en cuenta:

- La clase que funciona de Activator debe indicarse en el manifiesto.
- El método start debe preparar los recursos que un bundle necesita, es decir, registrar los servicios que proporciona y arrancar los threads pertinentes.
- El método stop debe deshacer lo que hizo el start para liberar los recursos. No hace falta desregistrar servicios porque el Framework se encarga de ello.

### 3.5.4 Servicios OSGi

En OSGi, las aplicaciones se construyen en torno a un conjunto de servicios cooperativos. Estos están disponibles a través del registro de servicios. Cada servicio queda definido semánticamente a través de la interfaz del servicio y un conjunto de propiedades, y es implementado por el objeto del servicio.

Un servicio debe contener los menores detalles de implementación posibles. De hecho, el estándar incluye un conjunto de servicios de uso común.

Cada servicio se ejecuta dentro y pertenece a un bundle en ejecución. Para ello, el bundle registra el servicio dentro del registro de servicios, junto con sus propiedades. Eventualmente, el bundle desregistrará el servicio, bien porque este parado o porque así lo desee. En cualquier caso, cuando un bundle es detenido, la plataforma ha de encargarse de desregistrar todos los servicios registrados por dicho bundle.

A nivel general OSGi tiene las siguientes capas, que se pueden observar también en la Ilustración 20. sobre estas capas se despliegan los diferentes servicios:

- Entorno de ejecución: define la especificación del entorno Java en el que se ejecutará el framework. Aunque la mayoría son válidos (J2SE, CDC, CLDC, MIDP,...) en OSGi también se define un entorno mínimo para los bundles OSGi.

- Capa de Módulos: define las políticas de carga de las clases,, basado en Java, pero añadiendo modularización.
- Capa de ciclo de vida: define el modelo de instalación, arranque, parada, actualización y desinstalación de los bundles.
- Capa de Service Registry: el Service Registry provee un modelo de cooperación entre los bundles, teniendo en cuenta el dinamismo de los mismos, y provee un mecanismo general para compartir objetos entre bundles. También están definidos un número de eventos (ServiceEvent, BundleEvent, FrameworkEvent) que anuncian el registro y desregistro de servicios.

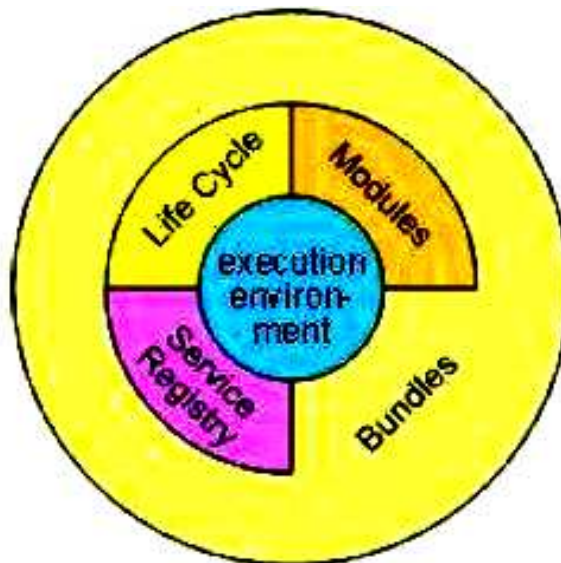


Ilustración 20. Capas de servicios OSGi

### 3.5.5 Implementaciones OSGi

Siendo OSGi una especificación abierta, cuenta con varias implementaciones, tanto libres como propietarias. Las implementaciones

libres tienen su código fuente disponible y son gratuitas, mientras que las propietarias el código fuente está oculto, y hay un coste por obtener sus licencias. Por eso en este proyecto, usaremos las implementaciones de código abierto:

- JEFFREE: (Java Embedded Framework FREE) Aplicación de código abierto para las pasarelas residenciales que quedó abierta la licencia en el 2003 y es para la especificación OSGi 2.0.
- OSCAR: (Open Service Container Architecture) Es una herramienta para generar paquetes, se denomina Mangel, para funcionalidades con HTTP y JMX.
- Knopflerfish: (antes Gamespace Telematic) Es una distribución libre de Knopflerfish OSGi, Makeware es el principal inversor y promotor de Knopflerfish, cumple con la especificación OSGi R4.
- Equinox: Desde un punto de vista de código, Equinox es una aplicación de la framework OSGi especificaciones R4, un conjunto de paquetes que implementan diversas funciones y servicios de OSGi para el funcionamiento de los sistemas basados en OSGi.
- Felix: Para el desarrollo del proyecto hemos decidido utilizar Felix. Sus orígenes fue un proyecto de la fundación Apache que se esfuerza por implementar una plataforma de servicio para la especificación R4 de OSGi, la cual incluye el Framework de OSGi y los servicios estándar además de proveer soporte para otras interesantes tecnologías relacionadas con OSGi. El objetivo final de Felix es ofrecer una implementación totalmente compatible del

Framework de OSGi y sus servicios estándares además de dar respaldo a toda la comunidad existente alrededor de esta tecnología. Actualmente, Felix implementa un gran parte de la versión de especificación completa. A pesar de este hecho, las funcionalidades que ofrece Felix del Framework de OSGi resultan muy estables. Además, debemos añadir que puede ser utilizado dentro de otros proyectos y utilizado como un plugin o mecanismo de extensión dinámico. Esta posibilidad la de a Felix mayor ventaja frente a otros sistemas utilizados para tareas similares, como el Java Management Extensions (JMX).

Por otro lado, las soluciones propietarias, son:

- Java Embedded Server (JES) de Sun microsystem.
- Service Manangment Framework (SMF) de IBM.
- mBedded Server de ProSyst.
- Espial DeviceTop 3.1 de Espial.
- aveLink Embedded Gateway de Antinav.
- Gatessapce Distributed Service Plataform (GDSP) de Gatespace AB.

## 3.6 *Interfaz Gráfica y Usabilidad*

El punto clave de este proyecto es la interfaz gráfica. En este capítulo explicaremos por qué es clave la interfaz gráfica en este tipo de desarrollos y también incluiremos un estudio de usabilidad.

### 3.6.1 **Interfaz Gráfica**

La interfaz de usuario es un componente crítico de la aplicación, ya que a través de ella, el usuario, interactúa con el sistema y para él el sistema es lo que ve y lo que toca. Una interfaz de usuario debe transmitir al usuario de la aplicación todo de lo que es capaz de hacer el sistema. El usuario sólo está interesado en la realización de unas tareas, para lo cual utiliza la aplicación, y ésta debe de poder realizar las tareas de la forma más rápida, intuitiva y eficaz posible. Por lo que una interfaz de usuario mal diseñada alcanza unos niveles de errores muy altos en la realización de tareas.

A continuación se muestran algunas estadísticas que reflejan la importancia que la interfaz de usuario tiene en el total del desarrollo de la aplicación [17]:

- La interfaz de usuario es aproximadamente el 60% de las líneas de código del total de un sistema de información interactivo.
- Una interfaz de usuario gráfica supone como mínimo el 29% del presupuesto de desarrollo de un sistema de información interactivo.

- El 80% de los costes en el ciclo de vida del desarrollo del software se producen después de que el sistema se haya liberado y a su vez el 80% de esos costes se deben a requisitos no cumplidos.

### 3.6.2 Usabilidad

El término usabilidad deriva del inglés, teniendo su origen en la expresión ‘user friendly’, que en castellano significa ‘facilidad de uso’. “La usabilidad es una característica que está relacionada con la medida de calidad de los sistemas interactivos usados por usuarios específicos en un contexto de uso, para conseguir objetivos específicos con efectividad, eficiencia y satisfacción, en lo relacionado a utilidad, facilidad de uso, facilidad de aprendizaje y apreciación” [17].

- Efectividad: la precisión y completitud con la que el usuario alcanza sus objetivos. Relacionados con este concepto está la facilidad de aprendizaje, la tasa de errores cometidos y la facilidad de recordar sus funcionalidades y procedimientos.
- Eficiencia: los recursos empleados relacionados con la precisión y completitud con que el usuario alcanza sus objetivos. Relacionados con este concepto está la facilidad de aprendizaje, la tasa de errores cometidos y la facilidad de la aplicación para ser recordada.
- Satisfacción: la ausencia de incomodidad y una actitud positiva hacia el uso de la aplicación. La aplicación debe ser agradable de usar de forma que el usuario esté satisfecho al utilizarla.



- Utilidad: capacidad de la aplicación para ayudar en la realización de tareas.
- Facilidad de uso: está relacionada con la eficiencia o efectividad con que se realizan las tareas. En una aplicación fácil de usar se realizarán las tareas más rápidamente.
- Facilidad de aprendizaje: es una medida del tiempo requerido para usar la aplicación con cierta eficiencia y llegar a recordar los procedimientos después de no usar la aplicación durante un tiempo determinado.
- Apreciación: es una medida de la percepción, opinión, sentimiento y actitud generada en el usuario por el uso de la aplicación. Es una medida subjetiva.

Los beneficios más importantes que aporta la usabilidad son:

- Incremento de la productividad de los usuarios de la aplicación, al reducir los tiempos para completar sus tareas y cometer menos errores que deberían ser corregidos mas tarde.
- Incremento del uso de la aplicación.
- Reducción del coste en documentación. Se necesitarán menos manuales de la aplicación.
- Reducción de los costes de soporte a la aplicación ya que resulta un producto fácil de instalar, de aprender y de usar.

- Reducción de los costes y tiempos de desarrollo, acertando a la primera con los requerimientos de los usuarios, ya que evita los rediseños y reduce el número de cambios posteriores en la aplicación.
- Detección temprana de fallos, aplicando usabilidad en las fases tempranas del desarrollo. Aunque en las primeras fases el coste de los errores es mucho menor.
- Reducción de los costes de mantenimiento de la aplicación.
- Aumento de la satisfacción del cliente reduciendo el esfuerzo de uso por parte del usuario y mejorando la calidad de vida de los usuarios.
- Aumento de la calidad del producto final con el consiguiente ahorro en control de calidad.
- Proporciona una ventaja competitiva para la organización. Lo más fácil de usar, es más fácil de vender.

En este proyecto para conseguir los objetivos de usabilidad, cumpliremos algunas de las normas básicas de usabilidad, en lo que tiene que ver con la presentación, los botones, los colores empleados, y la documentación mostrada al usuario durante la ejecución de la aplicación [18] [19].

- Presentación: Se debe crear un estilo de presentación (disposición y apariencia), que sea homogéneo en todas las pantallas de la aplicación.

- Botones: Los botones deben estar en una posición tal que sea fácil localizarlos y activarlos. Deben ser suficientemente grandes, por lo menos 7 mm x 7mm de superficie. Además, los botones de acciones opuestas deben estar separados y bien diferenciados. Y sus etiquetas imprescindibles para el manejo del producto deben ser fácilmente legibles.
- Colores: Las combinaciones del color de fondo y del primer plano debe contrastar lo suficiente, para que la imagen pueda ser visualizada de forma correcta por una persona con una deficiencia de percepción del color, o utilice una pantalla monocroma.
- Documentación: La documentación del producto debe estar redactada de la forma más clara y sencilla posible, con un vocabulario adecuado para la tarea realizada por el producto.

### ***3.7 Herramientas y tecnologías usadas***

Para nuestro Proyecto Fin de Carrera hemos necesitado emplear herramientas y tecnologías ya empleadas durante el estudio de la carrera, pro tanto, conocíamos ya la perfección, y las explicaremos, simplemente, a modo de referencia.

El lenguaje de programación elegido es JAVA en su versión “JSE6”. Esta versión ofrece mayor madurez, calidad, estabilidad, compatibilidad, escalabilidad y seguridad de la implementación de Java. Cuenta con gran número de funciones y mejoras, como el mejor funcionamiento de la interfaz gráfica y del control de las aplicaciones para interfaz gráfica,

además de mejoras y nuevas funciones de núcleo de servidor y de Java. Además, mantiene la compatibilidad con versiones anteriores garantizando la vigencia de las habilidades de los desarrolladores actuales de Java [20].

Como entorno de desarrollo se ha utilizado Eclipse. Se trata de un entorno de desarrollo integrado de código abierto multiplataforma. Uno de los entornos de desarrollo integrados frecuentemente usado es el IDE de JAVA, conocido como Java Development Toolkit (JDT). Comenzó como un proyecto de IBM en el año 2001, y actualmente es desarrollado por la Fundación Eclipse, que es una organización independiente que propone una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios [21].

Para la gestión de perfiles y dada la complejidad de los mismos, hemos empleado el lenguaje XML, ya que se trata de una tecnología que permite compartir los datos a todos los niveles entre todas las aplicaciones y soportes, posibilitando que si una aplicación escribe un documento de texto en formato XML, otra aplicación pueda recibirlo y procesarlo. Nació con el objetivo de solucionar las carencias presentadas por el lenguaje HTML, y orientada a Internet. Además de asegurar la compatibilidad entre diferentes sistemas de aplicación, se trata de un lenguaje extensible y fácil de programar y procesar [22].

Para describir la estructura del sistema, representar los diagramas de secuencia dentro del mismo y los diagramas de interacción del usuario con la aplicación, se han realizado diagramas de clases, de secuencia y casos de uso, respectivamente, utilizando para ello el lenguaje UML (Unified Modeling Language). Es un lenguaje de modelado visual que se usa para



especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. Es el lenguaje en el que está descrito el modelo. El lenguaje UML, que actualmente se encuentra respaldado por el OMG (Object Management Group) cuenta con diferentes tipos de diagramas, de los cuales nosotros hemos usado los diagramas de clase y los de casos de uso [23].



# *4 Estudio de Viabilidad y Presupuesto*

---

## *4.1 Estudio de Viabilidad*

Todo plan de viabilidad de un sistema tiene como objetivo tomar una decisión de si se deben poner en marcha el proyecto en el cual nos embarcamos, o por otro lado comprobar que la idea no resulta atrayente y será mejor olvidarla.

#### **4.1.1 La idea**

La idea de negocio es realizar una interfaz gráfica que permita gestionar los dispositivos domóticos instalados en la vivienda donde se está ejecutando la aplicación. Como valor añadido, se va a dotar a la interfaz de dos características que la distinguan de los demás productos existente sen el mercado:

- La interfaz será dinámica, permitiendo instalar nuevos dispositivos en la vivienda (o desinstalar) y que la interfaz los reconozca.
- La interfaz será genérica, no se limita a una sola vivienda, ya que el administrador del sistema deberá asignar una localización a los dispositivos, y las posibles localizaciones que tendrá para elegir el administrador dependerán de la vivienda en la que se encuentre.

A esto hay que añadir, que además ofrece interoperabilidad entre diferentes fabricantes.

#### **4.1.2 Análisis del entorno**

No nos podemos olvidar del entorno social y económico que nos encontramos. La ralentización de las ventas y el incremento de la competencia entre las promotoras llevan a éstas a buscar nuevas formas de atraer a los compradores. Por eso se encuentra en la domótica, un valor



diferenciador, que aporta la posibilidad de ofrecer un producto con más y nuevas prestaciones.

Ante la actual crisis económica que atraviesa el país, el proyecto que vamos a presentar al mercado, es idóneo para el ahorro en costes, como justifican las dos características marcadas en el aparatado anterior de este capítulo

Además, ofrecer interoperabilidad entre diferentes fabricantes, punto importante ya que como explicamos en la sección 3.1 el mercado de la domótica evoluciona lenta pero favorablemente gracias, entre otros factores, a la estandarización de la tecnología.

#### **4.1.3 Conclusiones**

Como vemos el sector necesita de nuestro producto por ser único, algo que no posee la competencia, algo innovador, ya que su gestión es dinámica permitiendo cambios en los dispositivos instalados sin tener que cambiar la interfaz, y no depende de la casa en la que se encuentre en ejecución. Además, ofrece interoperabilidad entre diferentes fabricantes, reconociendo cualquier dispositivo siempre y cuando su *bundle* instalado en la pasarela residencial registre el servicio DomoDevice; y también es indiferente del protocolo de comunicación que utilice el dispositivo, aunque en este proyecto no lo hemos implementado.

La única característica negativa, es que el despegue de inversión inicial es alto, ya que el coste de las pruebas de software se llevará gran parte de la inversión, al igual que la de promoción del producto añadido con la red de

comerciales. También, influirá negativamente, que no se trate de un producto terminado que se pueda ofrecer a los usuarios finales, y necesite de una segunda versión que permita la comunicación física con los dispositivos a través de un determinado protocolo de comunicación para poder vender a los usuarios.

## 4.2 *Presupuesto*

Este cálculo se ha realizado en función de las horas dedicadas a cada actividad, reflejadas en el apartado 1.1 Planificación del Proyecto, y los recursos consumidos en cada una de ella. Fijaremos que el salario medio de un ingeniero técnico de telecomunicaciones recién titulado en 8€/hora, y la jornada laboral constará de 7 horas.

La siguiente tabla muestra el presupuesto de cada actividad:

ID	Nombre	Duración	Presupuesto
1	Estudio de investigación	30 días	1680 €
2	Determinar viabilidad	2 días	112 €
3	Análisis y diseño	12 días	672 €
4	Implementación	24 días	1344 €

---

5	Pruebas	2 días	112 €
6	Documentación (*)	80 días	560 €

**Tabla 3. Presupuesto por fases**

(\*) Para calcular el presupuesto de la fase de documentación se ha tenido en cuenta que durante la elaboración de cada fase se ha ido generando documentación incluida en este documento. Por tanto se ha estimado, que después de realizar todas las actividades anteriores (72 días) la elaboración final de la memoria tendrá una duración de 10 días.

Como se puede comprobar el proyecto tendrá una duración aproximada de 4 meses (80 días teniendo en cuenta la semana laboral de Lunes a Viernes).

Para obtener el presupuesto total, además hay que tener en cuenta los costes directos y los costes indirectos:

- Costes directos: Estos costes serán los originados por la compra de elementos necesarios para el proyecto, así como formación o contratación de personal específico. En nuestro caso vendrá dado por la adquisición de un ordenador para su desarrollo. Este PC se ha utilizado durante la elaboración del proyecto (4 meses), por tanto, considerando que la máquina es amortizable en 2 años, su coste es  $1PC \cdot 1000 \text{ €} \cdot 4 \text{ meses} / (2 \text{ años} \cdot 12 \text{ meses}) = 166,66 \text{ €}$

- Costes indirectos: Estos costes serán los producidos por las licencias de herramientas genéricas de software, gracias a utilizar programas de código libre, como java para la programación orientada a objetos, dispondremos de un ahorro notable, otro término son costes ineludibles para la realización del proyecto como la luz, conexión a banda ancha y otros pagos regulares, estos podemos evaluarlos que supondrán un coste del 5% del coste total del proyecto.

Por tanto el presupuesto total del proyecto es:

**Presupuesto Total:**  $4.480 \text{ €} + 166,66 \text{ €} + 5 \% * 4.480 \text{ €} = 4.870,66 \text{ €}$

# *5 Análisis y Diseño del Sistema*

---

## **5.1 Definición del entorno**

Antes de empezar con el análisis y detallar el diseño de la aplicación, hay que concretar el contexto en el que se va a encontrar, es decir, definir el entorno de aplicación.

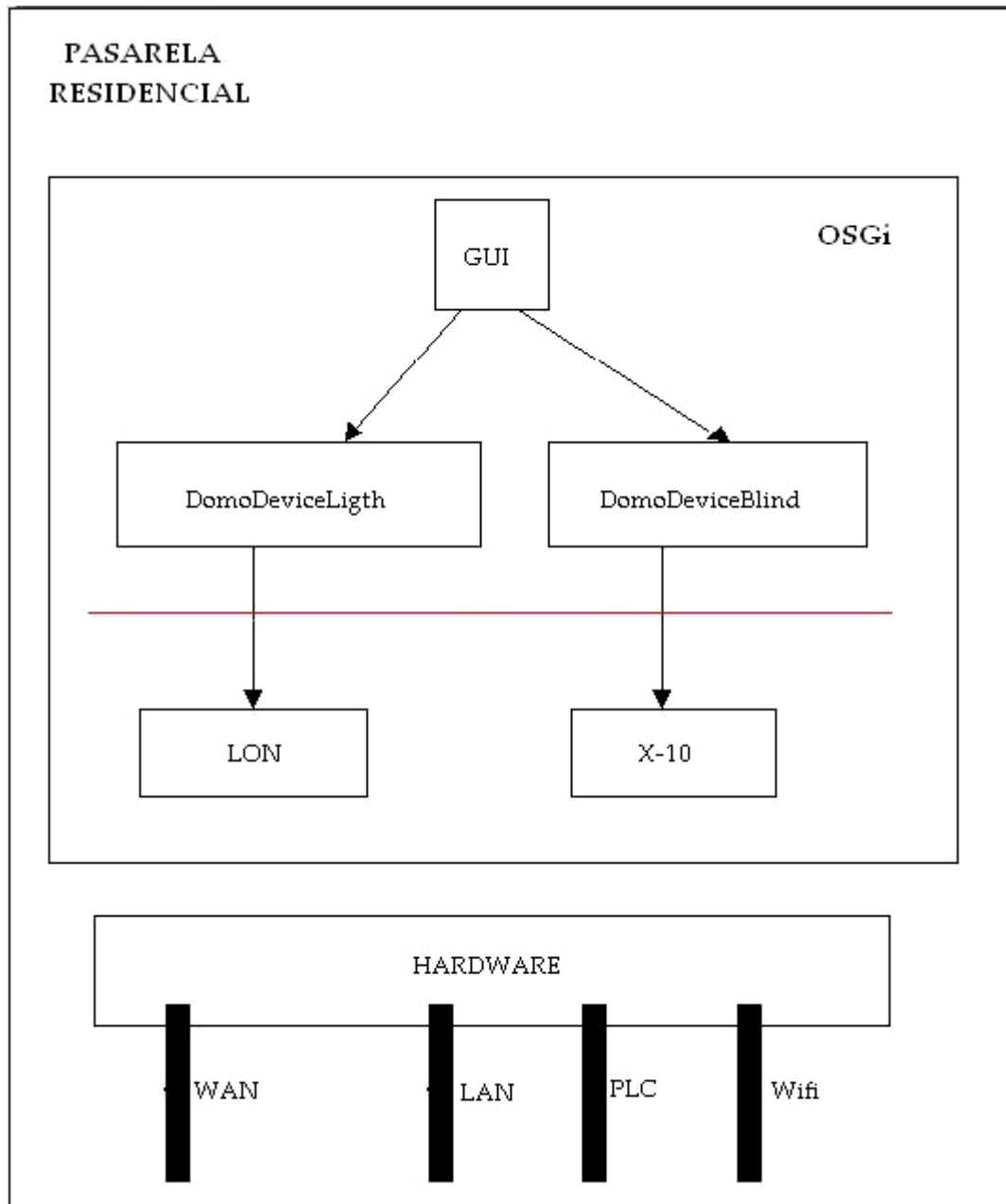


Ilustración 21. Definición del entorno

### 5.1.1 Pasarela Residencial

La Pasarela Residencial es el dispositivo central de control de la red residencial y sus servicios. Centraliza el control de los dispositivos

domóticos mediante software instalado sobre la plataforma OSGi que alberga la pasarela. Ese software se apoya sobre el hardware de dicho dispositivo, el cual proporciona diferentes interfaces de comunicación, como pueden ser Ethernet, PLC, o wifi, se empleará uno u otro en función del protocolo empleado (Lonworks, X-10, ect).

Como se puede apreciar en la Ilustración 21, en el entorno OSGi se encuentran instalados:

- El bundle de la interfaz, GUI: es el que recoge todos los servicios DomoDevice que anteriormente han registrado los dispositivos domóticos, y representa la interfaz gráfica de la aplicación.
- El bundle de los dispositivos, DomoDeviceLigth...: es el bundle que registra el servicio DomoDevice definido en esta aplicación.
- Y los bundles de los protocolos de control que se utilicen, Lonworks o X-10: Son los bundles encargados de comunicarse físicamente con los dispositivos. En este proyecto no se han implementado, se dejará para una segunda versión.

Conviene señalar que este proyecto sólo se ocupa de establecer el intercambio de información entre la interfaz gráfica y los dispositivos, a través de bundles que registran un servicio. Servicio que es definido en este propio proyecto (DomoDevice). De la forma que se ha definido este servicio, todo dispositivo instalado en la vivienda debe tener su bundle en la pasarela residencial.

### 5.1.2 Modelo de Negocio

Como acabamos de explicar, todo dispositivo instalado en la vivienda debe tener su bundle que registra el servicio DomoDevice instalado en la pasarela residencial para que la interfaz sea capaz de reconocerlo.

Esto quiere decir, que cuando el usuario realice la compra de un dispositivo y lo instale en su vivienda, necesita que el propio instalador, o el administrador del sistema, instale el bundle de dicho dispositivo. De esta forma, la interfaz automáticamente lo reconocerá. Dependiendo de lo potentes que sean los dispositivos, la instalación del bundle se podrá realizar automáticamente, incluyendo el propio dispositivo su bundle en la pasarela, o descargándolo de un enlace web... Aún así, después de instalar el dispositivo y que la interfaz lo reconozca, para que el usuario pueda interactuar con él, el administrador deberá agregarlo a la interfaz, asignándole un nombre y una localización específica. Aunque también, podría ser posible que el administrador accediera a la aplicación de manera remota.

El objetivo final es que el usuario compre su dispositivo y no tenga que realizar más tareas para tenerlo instalado en la interfaz, consiguiendo así una aplicación lo más dinámica y genérica posible.

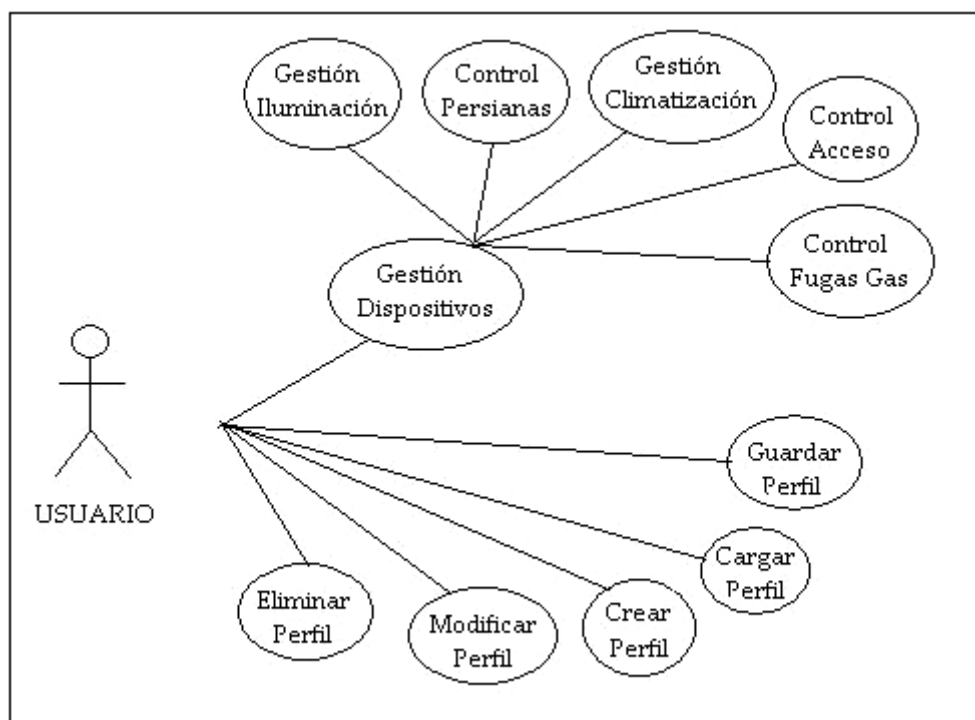
## 5.2 *Casos de Uso*

Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso nos

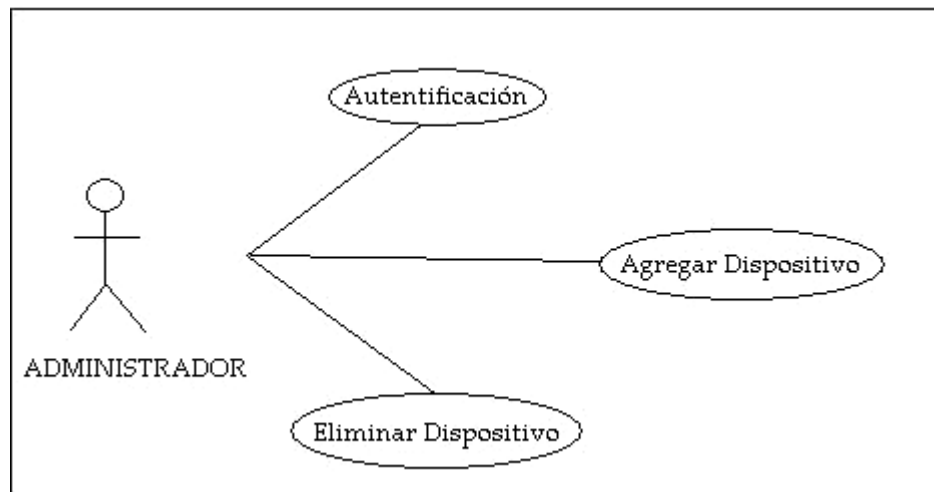


proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario para conseguir un objetivo específico. En definitiva, sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios.

En este capítulo, describiremos cada uno de los casos de uso para nuestra aplicación, y representaremos el conjunto en dos diagramas, ya que diferenciamos el tipo de usuario que realiza la interacción, por un lado está el usuario final, referenciado como “Usuario”, y por otro el “Administrador” del sistema.



**Ilustración 22. Diagrama de Casos de Uso del Usuario**



**Ilustración 23. Diagrama de Casos de Uso del Administrador**

Para describir los casos de uso de nuestra aplicación, se emplearán tabas auto explicativas como la del siguiente ejemplo:

Identificador	Caso de Uso de Ejemplo
<b>Descripción</b>	Descripción del requisito
<b>Actores</b>	El tipo de usuario que interacciona con el sistema
<b>Objetivo</b>	Servicio que el actor busca conseguir
<b>Pre- Condiciones</b>	Descripción del conjunto de estados del sistema anteriores a la ejecución del caso de uso
<b>Post- Condiciones</b>	Descripción del conjunto de estados del sistema anteriores a la ejecución del caso de uso
<b>Escenario</b>	Secuencia de acciones principales (información intercambiada en la interacción del escenario básico)

**Tabla 4. Caso de uso de ejemplo**

Ahora describiremos cada caso de uso por separado atendiendo a los actores, el objetivo u objetivos a cumplir, las pre-condiciones que se tienen

que dar antes de la ejecución del caso de uso, las post-condiciones que reflejan el estado del sistema después de la ejecución del caso de uso, y el escenario en el que tiene lugar la interacción entre el usuario y el sistema.

Cada caso de uso tendrá asociado un identificador que permita diferenciarlo del resto de casos de usos. El identificador tendrá la siguiente estructura:

- Caso de uso (CU): Las dos primeras letras serán comunes para todos ya que identifican que son casos de uso, CU.
- Tipo de Usuario: La tercera letra identifica al tipo de usuario que pertenece el caso de uso, siendo U para el usuario final (habitante de la vivienda), y A para el administrador.
- Número de caso de uso: Indica el número del caso de uso dentro de la clasificación de su tipo de usuario.

Ejemplo:

#### **CU-U01: Caso De Uso del Usuario número 1**

Diferenciaremos dos tipos de usuarios posibles para el sistema, así trataremos por un lado al usuario tratado como si fuese el habitante de la vivienda (que constituye el usuario final), y por otra parte al administrador del sistema. De esta forma los dos tipos de usuarios son: Usuario y Administrador.

### 5.2.1 Casos de Uso del Usuario

CU-U1	Gestión de Dispositivos
<b>Descripción</b>	Engloba todas las interacciones que el usuario tiene con los dispositivos domóticos instalados en la vivienda
<b>Actores</b>	Los habitantes de la vivienda
<b>Objetivo</b>	Modificar el estado de cualquier dispositivo instalado en el hogar
<b>Pre- Condiciones</b>	Que la aplicación se encuentre en la pantalla de 'Gestión Dispositivos'
<b>Post- Condiciones</b>	El dispositivo cambia de estado.
<b>Escenario</b>	El usuario pulsa sobre el botón 'Gestión Dispositivos' del menú principal. A continuación, selecciona un dispositivo y a la derecha se le muestran los distintos botones que permiten ejecutar una acción sobre ese dispositivo. El Usuario deberá pulsar sobre el botón deseado para ejecutar la acción. El sistema muestra una pantalla confirmando el cambio de estado del dispositivo

Tabla 5. CU-U1, Gestión Dispositivos

CU-U2	Gestión de la Iluminación
<b>Descripción</b>	Encender o apagar las luces de la vivienda

<b>Actores</b>	Los habitantes de la vivienda
<b>Objetivo</b>	Modificar el estado de la iluminación en cualquier dispositivo de luz instalado en el hogar
<b>Pre- Condiciones</b>	Que la aplicación se encuentre en la pantalla de 'Gestión Dispositivos'
<b>Post- Condiciones</b>	El estado del dispositivo de la luz cambia a encendido o apagado según la acción. O si se trata de un dispositivo con capacidad de regular la intensidad luminosa, el usuario habrá variado la intensidad de luz
<b>Escenario</b>	El usuario pulsa sobre el botón 'Gestión Dispositivos' del menú principal. A continuación, selecciona el dispositivo de luz adecuado y a la derecha se le muestran los distintos botones que permiten ejecutar una acción sobre el dispositivo. El Usuario deberá pulsar sobre el botón deseado para ejecutar la acción. El sistema muestra una pantalla confirmando el cambio de estado del dispositivo

**Tabla 6.CU-U2, Gestión de la Iluminación**

<b>CU-U3</b>	<b>Gestión de las Persianas</b>
<b>Descripción</b>	Subir o bajar las persianas de la vivienda
<b>Actores</b>	Los habitantes de la vivienda



<b>Objetivo</b>	Modificar el estado de las persianas del hogar
<b>Pre- Condiciones</b>	Que la aplicación se encuentre en la pantalla de 'Gestión Dispositivos'
<b>Post- Condiciones</b>	El estado de la persiana es modificado. La persiana estará subida, bajada, y también podrá modificarse en base a unos niveles
<b>Escenario</b>	El usuario pulsa sobre el botón 'Gestión Dispositivos' del menú principal, para acceder al panel que permita gestionar el estado de cada dispositivo. A continuación, selecciona el dispositivo de persiana adecuado y a la derecha se le muestran los distintos botones que permiten ejecutar una acción sobre el dispositivo. El Usuario deberá pulsar sobre el botón deseado para ejecutar la acción. El sistema muestra una pantalla confirmando el cambio de estado del dispositivo

Tabla 7.CU-U3, Gestión de las Persianas

<b>CU-U4</b>	<b>Gestión de la Climatización</b>
<b>Descripción</b>	Subir o bajar la temperatura de una habitación de la vivienda
<b>Actores</b>	Los habitantes de la vivienda
<b>Objetivo</b>	Modificar el estado de la temperatura en cualquier sala de la casa



<b>Pre- Condiciones</b>	Que la aplicación se encuentre en la pantalla de 'Gestión Dispositivos'
<b>Post- Condiciones</b>	La temperatura en la sala en la que se encuentra el dispositivo seleccionado ha cambiado
<b>Escenario</b>	El usuario pulsa sobre el botón 'Gestión Dispositivos' del menú principal, para acceder al panel que permita gestionar el estado de cada dispositivo. A continuación, selecciona el dispositivo de aire acondicionado / calefacción adecuado y a la derecha se le muestran los distintos botones que permiten ejecutar una acción sobre el dispositivo. El Usuario deberá pulsar sobre el botón deseado para ejecutar la acción. El sistema muestra una pantalla confirmando el cambio de estado del dispositivo

Tabla 8.CU-U4, Gestión de la Climatización

<b>CU-U5</b>	<b>Control de Acceso</b>
<b>Descripción</b>	Abrir o cerrar la puerta de acceso a la vivienda
<b>Actores</b>	Los habitantes de la vivienda
<b>Objetivo</b>	Control el acceso de personas a la vivienda
<b>Pre- Condiciones</b>	Que la aplicación se encuentre en la pantalla de 'Gestión Dispositivos'
<b>Post- Condiciones</b>	El estado de la puerta de acceso a la vivienda habrá cambiado



<b>Escenario</b>	El usuario pulsa sobre el botón 'Gestión Dispositivos' del menú principal, para acceder al panel que permita gestionar el estado de cada dispositivo. A continuación, selecciona el dispositivo de la puerta y a la derecha se le muestran los distintos botones que permiten ejecutar una acción sobre el dispositivo. El Usuario deberá pulsar sobre el botón deseado para ejecutar la acción. El sistema muestra una pantalla confirmando el cambio de estado del dispositivo
------------------	--

Tabla 9.CU-U5, Control de Acceso

<b>CU-U6</b>	<b>Control de las Fugas de Gas</b>
<b>Descripción</b>	Abrir o cerrar la compuerta de suministro de gas en la vivienda
<b>Actores</b>	Los habitantes de la vivienda
<b>Objetivo</b>	Cortar o abrir el suministro de gas en la vivienda.
<b>Pre- Condiciones</b>	Que la aplicación se encuentre en la pantalla de 'Gestión Dispositivos'
<b>Post- Condiciones</b>	La compuerta de gas ha modificado su posición



<b>Escenario</b>	<p>El usuario pulsa sobre el botón ‘Gestión Dispositivos’ del menú principal, para acceder al panel que permita gestionar el estado de cada dispositivo. A continuación, selecciona el dispositivo de llave de gas adecuado y a la derecha se le muestran los distintos botones que permiten ejecutar una acción sobre el dispositivo. El Usuario deberá pulsar sobre el botón deseado para ejecutar la acción. El sistema muestra una pantalla confirmando el cambio de estado del dispositivo</p>
------------------	---

**Tabla 10.CU-U6, Control de las Fugas de Gas**

<b>CU-U7</b>	<b>Guardar Perfil</b>
<b>Descripción</b>	Almacenar la configuración actual de los dispositivos
<b>Actores</b>	Los habitantes de la vivienda
<b>Objetivo</b>	Almacenar una configuración específica, en un perfil, de todos los dispositivos, para después poder cargarla y establecer los dispositivos según se guardaron en dicho perfil
<b>Pre- Condiciones</b>	Que la aplicación se encuentre en la pantalla de ‘Gestión Dispositivos’
<b>Post- Condiciones</b>	Se habrá creado un perfil que almacene una configuración específica de cada dispositivo

<b>Escenario</b>	<p>El usuario en el menú principal de la aplicación selecciona el botón 'Gestión Dispositivos', y después de configurar los dispositivos pulsa el botón 'Grabar configuración' que se encuentra en la esquina inferior derecha de la pantalla. A continuación, aparece una ventana pequeña que le pide al usuario que introduzca un nombre para asignar al perfil. El usuario introduce el nombre y pulsa el botón OK Si el nombre no existía lo almacenará directamente; pero si, por el contrario, el nombre introducido ya corresponde a otro perfil la aplicación pregunta al usuario si desea sobrescribir el perfil o desea asignarle otro nombre</p>
------------------	---

Tabla 11.CU-U7, Guardar Perfil

CU-U8	Crear Perfil
<b>Descripción</b>	Almacenar la configuración actual de los dispositivos
<b>Actores</b>	Los habitantes de la vivienda
<b>Objetivo</b>	Almacenar una configuración específica, en un perfil nuevo, de todos los dispositivos, para después poder cargarla y establecer los dispositivos según se guardaron en dicho perfil
<b>Pre- Condiciones</b>	Que la aplicación se encuentre en la pantalla de 'Gestión Perfiles'
<b>Post- Condiciones</b>	Se habrá creado un perfil que almacene una configuración específica de cada dispositivo

<b>Escenario</b>	<p>El usuario en el menú principal pulse el botón ‘Gestión Perfiles’ y en ese panel elija la opción ‘Crear Perfil’. La aplicación le dirigirá hacia la pantalla de ‘Gestión Dispositivos’, y después de configurar los dispositivos pulsa el botón ‘Grabar configuración’ que se encuentra en la esquina inferior derecha de la pantalla. A continuación, aparece una ventana pequeña que le pide al usuario que introduzca un nombre para asignar al perfil. El usuario introduce el nombre y pulsa el botón OK Si el nombre no existía lo almacenará directamente; pero si, por el contrario, el nombre introducido ya corresponde a otro perfil la aplicación pregunta al usuario si desea sobrescribir el perfil o desea asignarle otro nombre</p>
------------------	--

**Tabla 12.CU-U8, Crear Perfil**

<b>CU-U9</b>	<b>Modificar Perfil</b>
<b>Descripción</b>	Modificar la información almacenada en un perfil de configuración
<b>Actores</b>	Los habitantes de la vivienda
<b>Objetivo</b>	Modificar un perfil anteriormente creado
<b>Pre- Condiciones</b>	Que la aplicación se encuentre en la pantalla de ‘Gestión Perfiles’
<b>Post- Condiciones</b>	Se habrá modificado un perfil

<b>Escenario</b>	<p>El usuario en el menú principal pulsa el botón ‘Gestión Perfiles’.</p> <p>En ese panel primero elige el perfil que desea modificar y después debe pulsar sobre el botón ‘Modificar Perfil’. La aplicación le dirigirá hacia la pantalla de ‘Gestión Dispositivos’ mostrando los dispositivos en la configuración guardada.</p> <p>Después de configurar de nuevo los dispositivos, el usuario deberá pulsar el botón ‘Grabar configuración’ que se encuentra en la esquina inferior derecha de la pantalla. A continuación, aparece una ventana pequeña que le pide al usuario que introduzca un nombre para asignar al perfil. Si el usuario desea sobrescribirlo deberá introducir el mismo nombre de perfil y cuando la aplicación le pregunte si desea sobrescribir, pulsar ‘Sí’. Si no desea sobrescribir deberá introducir un nombre de perfil que no exista.</p>
------------------	--

**Tabla 13.CU-U9, Modificar Perfil**

<b>CU-U10</b>	<b>Cargar Perfil</b>
<b>Descripción</b>	Cargar en la aplicación la información almacenada en un perfil
<b>Actores</b>	Los habitantes de la vivienda
<b>Objetivo</b>	Cargar un perfil anteriormente creado
<b>Pre- Condiciones</b>	Que la aplicación se encuentre en la pantalla de ‘Gestión Perfiles’

<b>Post- Condiciones</b>	Los dispositivos han cambiado su estado, adquiriendo la configuración que estaba almacenada en el perfil
<b>Escenario</b>	El usuario en el menú principal pulsa el botón 'Gestión Perfiles'. En ese panel primero elige el perfil que desea modificar y después debe pulsar sobre el botón 'Cargar Perfil'. La aplicación le dirigirá hacia la pantalla de 'Gestión Dispositivos' mostrando los dispositivos en la configuración guardada en el perfil que se ha cargado

**Tabla 14.CU-U10, Cargar Perfil**

<b>CU-U11</b>	<b>Eliminar Perfil</b>
<b>Descripción</b>	Eliminar un perfil
<b>Actores</b>	Los habitantes de la vivienda
<b>Objetivo</b>	Eliminar un perfil anteriormente creado
<b>Pre- Condiciones</b>	Que la aplicación se encuentre en la pantalla de 'Gestión Perfiles'
<b>Post- Condiciones</b>	Se ha eliminado un archivo que almacenaba un perfil

<b>Escenario</b>	El usuario en el menú principal pulsa el botón 'Gestión Perfiles'. En ese panel primero elige el perfil que desea modificar y después debe pulsar sobre el botón 'Eliminar Perfil'. La aplicación mostrará una ventana de seguridad para que el usuario confirme que realmente desea eliminar ese perfil seleccionado
------------------	---

Tabla 15.CU-U11, Eliminar Perfil

### 5.2.2 Casos de Uso del Administrador

<b>CU-A1</b>	<b>Autenticación</b>
<b>Descripción</b>	El administrador se identifica como tal
<b>Actores</b>	El administrador del sistema
<b>Objetivo</b>	Restricción del acceso al sistema
<b>Pre- Condiciones</b>	Que la pestaña seleccionada de la aplicación sea 'Administrador' dentro del panel 'Gestión Dispositivos'
<b>Post- Condiciones</b>	El administrador accede a la aplicación para poder realizar sus tareas encomendadas
<b>Escenario</b>	El administrador selecciona la pestaña 'Administrador' dentro del panel 'Gestión Dispositivos'. Aparecerá una ventana pequeña en la que el administrador deberá rellenar el campo de Nombre y Contraseña para poder acceder, y pulsar el botón 'OK'

Tabla 16.CU-A1, Autenticación

CU-A2	Agregar Dispositivos
Descripción	Agregar dispositivos a la vivienda, asignándoles un nombre y una localización concreta dentro de la vivienda
Actores	El administrador del sistema
Objetivo	Añadir dispositivos a la aplicación, para que el usuario pueda interaccionar con ellos
Pre- Condiciones	El administrador ha debido autenticarse
Post- Condiciones	La aplicación cuenta con nuevos dispositivos añadidos para que después el usuario pueda interaccionar con ellos
Escenario	Después de autenticarse, el administrador seleccionará aquél o aquellos dispositivos que necesita añadir, después elegirá una determinada ubicación para el dispositivo en la vivienda y le asignará un nombre para que luego sea identificado por el usuario. A continuación, pulsará el botón 'Agregar Dispositivo'. A la derecha se muestra un texto que indica los dispositivos que ya están añadidos.

Tabla 17.CU-A2, Agregar Dispositivos

CU-A3	Eliminar Dispositivos
Descripción	Eliminar dispositivos de la vivienda
Actores	El administrador del sistema

<b>Objetivo</b>	Eliminar dispositivos de la aplicación
<b>Pre- Condiciones</b>	El administrador ha debido autenticarse
<b>Post- Condiciones</b>	El número de dispositivos con los que cuenta la aplicación para que después el usuario pueda interaccionar con ellos, ha disminuido
<b>Escenario</b>	Después de autenticarse, el administrador seleccionará aquél o aquellos dispositivos que necesita eliminar. A continuación, pulsará el botón 'Eliminar Dispositivo'. A la derecha se muestra un texto que indica los dispositivos que están añadidos.

Tabla 18.CU-A3, Eliminar Dispositivos

## 5.3 *Especificaciones de requisitos de software*

En este capítulo se ofrece una descripción completa de cada requisito que debe cumplir la aplicación para el correcto funcionamiento del sistema. Como requisito se entiende “la característica del sistema que es una condición para su aceptación por el cliente” [24].

Cada requisito tendrá asociado un identificador que permita diferenciarlo del resto de requisitos. El identificador tendrá la siguiente estructura:



- Tipo de requisito (RXX): Según el caso será de dos o de tres letras. Para indicar que es un requisito de tipo funcional, será RF. Y para indicar que es de tipo no funcional, será RNF.
- Número de requisito: Indica el número del requisito dentro de la clasificación de su tipo.

Ejemplo:

#### **RF-01: Requisito Funcional número 1**

Existen dos tipos diferentes de requisitos. En este proyecto distinguiremos entre requisitos funcionales y requisitos no funcionales. Los requisitos funcionales responden a los casos de uso, y los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación.

### **5.3.1 Requisitos Funcionales**

<b>RF-01</b>	<b>Gestión Iluminación</b>
<b>Descripción</b>	El usuario puede gestionar y controlar el estado de los dispositivos de luz
<b>Prioridad</b>	Alta
<b>Comentarios</b>	Ninguno

**Tabla 19.RF-01, Gestión Iluminación**



RF-02	Gestión Persianas
Descripción	El usuario puede gestionar y controlar el estado de los dispositivos de luz
Prioridad	Alta
Comentarios	Ninguno

Tabla 20.RF-02, Gestión Persianas

RF-03	Gestión Temperatura
Descripción	El usuario puede gestionar y controlar el estado de la temperatura de cada habitación
Prioridad	Alta
Comentarios	Ninguno

Tabla 21.RF-03, Gestión Temperatura

RF-04	Gestión Fugas de Gas
Descripción	El usuario puede gestionar y controlar el suministro de gas en la vivienda
Prioridad	Alta
Comentarios	Normalmente podrá ir asociado a un sensor, que en caso de fuga, cierre la compuerta de gas automáticamente

Tabla 22.RF-04, Gestión Fugas de Gas



RF-05	Gestión Acceso
Descripción	El usuario puede gestionar y controlar el acceso a la vivienda
Prioridad	Alta
Comentarios	Normalmente podrá ir asociado a un sensor, que en caso de robo, active una alarma

Tabla 23.RF-05, Gestión Acceso

RF-06	Guardar Perfil
Descripción	El usuario puede almacenar el estado concreto de todos los dispositivos instalados en el hogar en un perfil
Prioridad	Alta
Comentarios	Ninguno

Tabla 24.RF-06, Guardar Perfil

RF-07	Botón Crear Perfil
Descripción	La aplicación dirige al usuario al panel Gestión Dispositivos
Prioridad	Alta
Comentarios	En el panel Gestión Dispositivos el usuario puede configurar a su gusto todos los dispositivos y posteriormente almacenarlos en un perfil nuevo

Tabla 25.RF-07, Botón Crear Perfil



RF-08	Modificar Perfil
Descripción	El usuario puede modificar esos perfiles que anteriormente creó
Prioridad	Media
Comentarios	Ninguno

Tabla 26.RF-08, Modificar Perfil

RF-09	Cargar Perfil
Descripción	El usuario puede cargar a un perfil que anteriormente creó para disponer de los dispositivos en el estado en el que fueron almacenados
Prioridad	Alta
Comentarios	Ninguno

Tabla 27.RF-09, Cargar Perfil

RF-10	Eliminar Perfil
Descripción	El usuario puede borrar los perfiles que anteriormente creó
Prioridad	Alta
Comentarios	Ninguno

Tabla 28.RF-10, Eliminar Perfil



RF-11	Botón Ayuda
Descripción	Tanto usuario como administrador pueden solicitar ayuda cuando no sepa qué hacer en un momento determinado mientras utiliza la aplicación
Prioridad	Alta
Comentarios	Ninguno

Tabla 29.RF-11, Botón Ayuda

RF-12	Aviso cambio de estado de dispositivo
Descripción	La aplicación avisará al usuario cuando éste modifique el estado de algún dispositivo domótico
Prioridad	Alta
Comentarios	Ninguno

Tabla 30.RF-12, Aviso cambio de estado de dispositivo

RF-13	Autenticarse
Descripción	El administrador puede autenticarse en la aplicación para acceder a su panel de información
Prioridad	Alta
Comentarios	Deberá introducir un nombre y un password

Tabla 31.RF-13, Autenticarse

RF-14	Añadir Dispositivos
Descripción	El administrador puede añadir dispositivos a la aplicación
Prioridad	Alta
Comentarios	Para añadirlos, también debe asignarles un nombre y una ubicación concreta

Tabla 32.RF-14, Añadir Dispositivos

RF-15	Eliminar Dispositivos
Descripción	El administrador puede eliminar dispositivos de la aplicación
Prioridad	Alta
Comentarios	Ninguno

Tabla 33.RF-15, Eliminar Dispositivos

RF-16	Interfaz dinámica
Descripción	Ante cambios en el número de dispositivos instalados, la interfaz debe actualizarse en caliente, mostrando en cada momento la información sobre los dispositivos que estén instalados en el hogar
Prioridad	Alta
Comentarios	Si algún dispositivo se desinstala directamente de la casa, o se estropea, o se instala un nuevo dispositivo en la vivienda, la información presentada debe actualizarse

Tabla 34.RF-16, Interfaz dinámica

### 5.3.2 Requisitos No Funcionales

RNF-01	Tiempo de respuesta
Descripción	El sistema debe ser rápido y el tiempo de respuesta debe ser el mínimo posible
Prioridad	Alta
Comentarios	Ninguno

Tabla 35.RNF-01, Tiempo de respuesta

RNF-02	Requisitos de equipos Físicos
Descripción	La aplicación necesitará una serie de requisitos de hardware, como son un PC de una velocidad de procesamiento de 3Ghz como mínimo, 512 MB de memoria RAM, y una resolución mínima de pantalla de 800x600
Prioridad	Alta
Comentarios	Ninguno

Tabla 36.RNF-02, Requisitos de equipos Físicos

RNF-03	Comprobación del paradigma
Descripción	Se comprobará que la aplicación se ha diseñado y desarrollado siguiendo la orientación a objetos y la tecnología OSGi
Prioridad	Alta
Comentarios	Ninguno

Tabla 37.RNF-03, Comprobación del paradigma

RNF-04	Entorno Java
Descripción	La aplicación funcionará en cualquier máquina que tenga instalada el entorno JSE versión 6
Prioridad	Alta
Comentarios	Ninguno

Tabla 38.RNF-04, Entorno Java

Para cumplir los objetivos de usabilidad marcados en el apartado 3.6.2 de este documento, se establecen los siguientes requisitos:

RNF-05	Presentación
Descripción	La aplicación sigue una estructuración lógica, y siempre constante durante toda su ejecución
Prioridad	Alta
Comentarios	Ninguno

Tabla 39.RNF-05, Presentación

RNF-06	Colores
Descripción	Los colores empleados deben ser de fácil distinción para evitar problemas ante usuarios con daltonismo
Prioridad	Alta
Comentarios	Ninguno

Tabla 40.RNF-06, Colores



RNF-07	Botones
Descripción	Los botones se encuentran separados y tiene un tamaño grande
Prioridad	Alta
Comentarios	Los botones de regresar al Menú Principal y de Ayuda se encuentran colocados siempre en la misma posición. Los botones de acción sobre los dispositivos se encuentran separados. Y el botón salir se encuentra en la pantalla principal de la aplicación en la parte superior izquierda, de forma que sea visible al usuario y esté alejado de los otros botones principales.

Tabla 41.RNF-07, Botones

RNF-08	Documentación
Descripción	La información se muestra al usuario de forma muy sencilla
Prioridad	Alta
Comentarios	Además de los textos mostrados en la pantalla indicando qué dispositivos están instalados, y su estado actual, en el botón ayuda se le informa al usuario de las posibilidades que le ofrece la aplicación

Tabla 42.RNF-08, Documentación

## 5.4 *Modelo Conceptual*

### 5.4.1 Diagrama de Clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones

entre ellos. La siguiente imagen muestra el diagrama de clases de la aplicación:

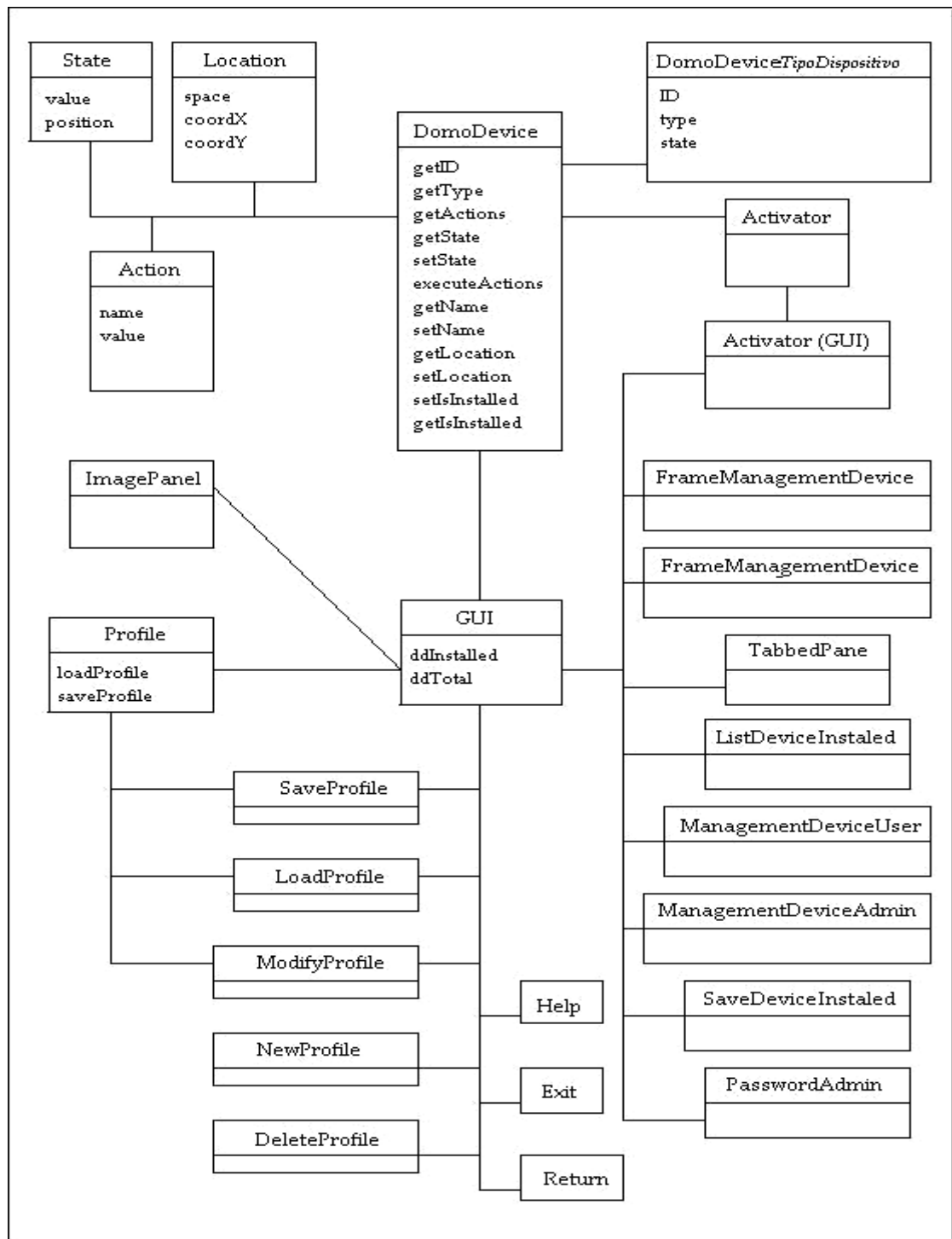


Ilustración 24. Diagrama de Clases

A continuación, se detalla el diagrama de clases conceptual que representa la Ilustración 24:

- **DomoDevice:** Esta clase es una interfaz que define el servicio proporcionado por todos los dispositivos, es decir, define todas las operaciones (métodos) que podrán realizar los dispositivos, los cuales implementarán todos esos métodos.
- **DomoDevice*TipoDispositivo*:** Esta clase representa las clases de cada dispositivo (DomoDeviceLigth, DomoDeviceBlind,...) Cada una de ellas implementa todos los métodos definidos por la interfaz DomoDevice. Como atributos especiales destacar el “ID” y el “type”, que almacenan el ID (único para cada dispositivo) y el tipo de dispositivo concreto, respectivamente. También destacar el método “execute”, que no está incluido en la interfaz “DomoDevice”, y que sirve para simular la comunicación entre el bundle del protocolo empleado en la vivienda (Lonworks, X-10,...) y el propio dispositivo, para mandar una determinada acción al dispositivo. En nuestro caso, simula esa comunicación imprimiendo un mensaje por pantalla.
- **Activator (dispositivos):** Todos los bundles de los dispositivos incluyen además de la clase “DomoDevice*TipoDispositivo*”, una clase “Activator”, que sirve para que el dispositivo registre el servicio DomoDevice.
- **State:** Esta clase proporciona la información sobre el estado en el que se encuentra cualquier dispositivo. Hay dos formas de asignar

un estado a un dispositivo: con un valor, o con un valor y una posición. Según sea necesario, se usará una u otra.

- Action: Esta clase se utiliza para simular, de cierta manera, lo que sería el intercambio de información entre el dispositivo y la red del protocolo de comunicación que posea la vivienda (Lonworks, X10,...). Almacenando la información de la acción en dos atributos, “name” y “value”, que corresponden al nombre que identifica la acción y el valor de la misma.
- Location: Esta clase proporciona la información sobre la localización en la que se encuentra cualquier dispositivo domótico. La localización se expresa con 3 atributos: “space”, “coordX” y “coordY”, que representan el espacio dentro de la casa (salón, cocina,...) y la coordenada X y coordenada Y concreta, que ocupa dentro de la casa, respectivamente.
- GUI: Esta clase representa la interfaz gráfica de la aplicación. Por la interacción del usuario, interactúa con otras clases para la gestión de datos (perfiles, estado de dispositivos,...). Sus atributos principales son “ddInstalled” y “ddTotal” con los que realiza toda la gestión de control.
- Activator (GUI): El bundle que representa la interfaz de la aplicación también incluye una clase “Activator”. En este caso, esta clase es la encargada de recoger todos los servicios que anteriormente han registrado los dispositivos. Además, comparará cada uno de esos dispositivos con los que estén agregados en la interfaz (o con los que estén contenidos dentro de

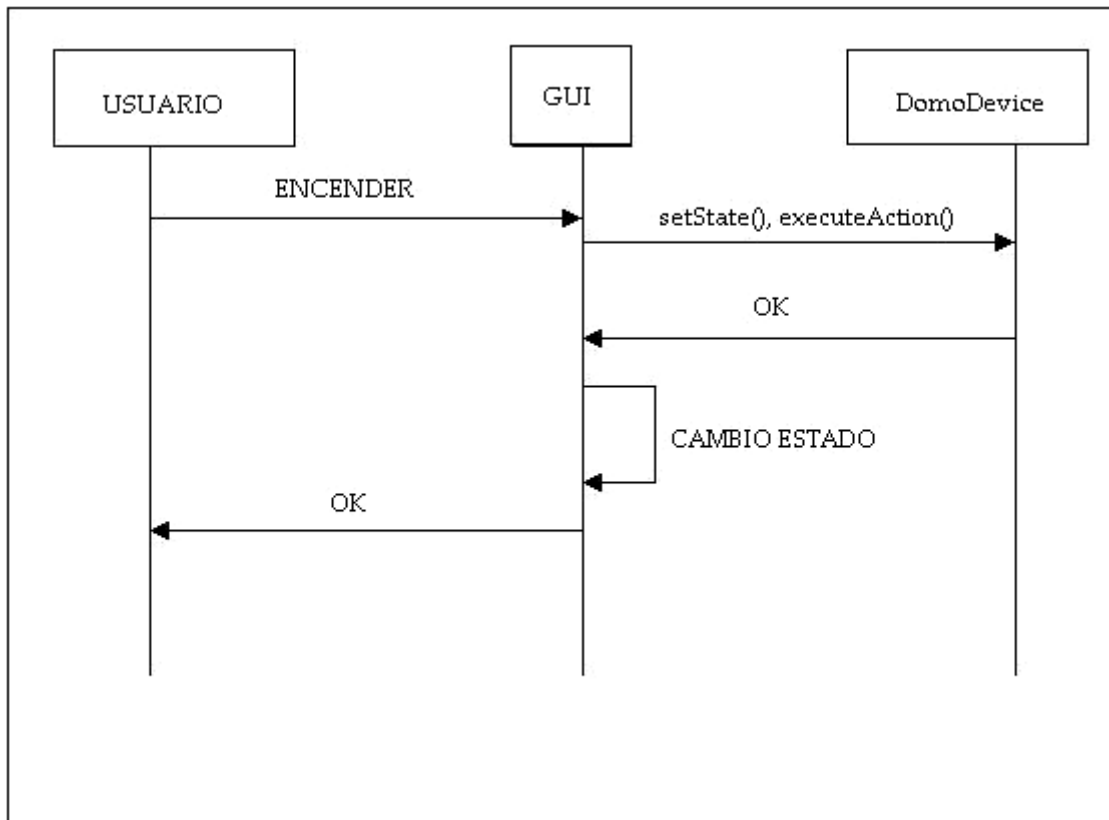
un perfil), para que la interfaz pueda presentar tanto todos los dispositivos que hayan registrado el servicio, como aquellos dispositivos que aparte de registrar el servicio, la interfaz ya tenga agregados.

- Profile: Esta clase permite gestionar los perfiles de configuración registrados en la vivienda, implementando las operaciones de guardar y cargar perfil, y el nombre de dicho perfil en el atributo “name”.
- ImagePanel: Esta clase es usada para poder representar imágenes de fondo, como la de la pantalla principal que da la bienvenida al usuario, y las imágenes que muestran el plano de la vivienda.
- El resto de clases representan las clases con las que interactúa la interfaz de la aplicación GUI, por la interacción del usuario con la aplicación. Por tanto, esas clases realizan el manejo de los eventos lanzados por el usuario. Todas ellas tienen como atributo “gui”, que representa la interfaz, para así poder tener acceso a todos los atributos y métodos incluidos en la clase “GUI”.

### **5.4.2 Diagramas de Secuencias**

Un diagrama de secuencia muestra la interacción entre los objetos de una aplicación a través del tiempo y se modela para cada método de la clase. Muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas

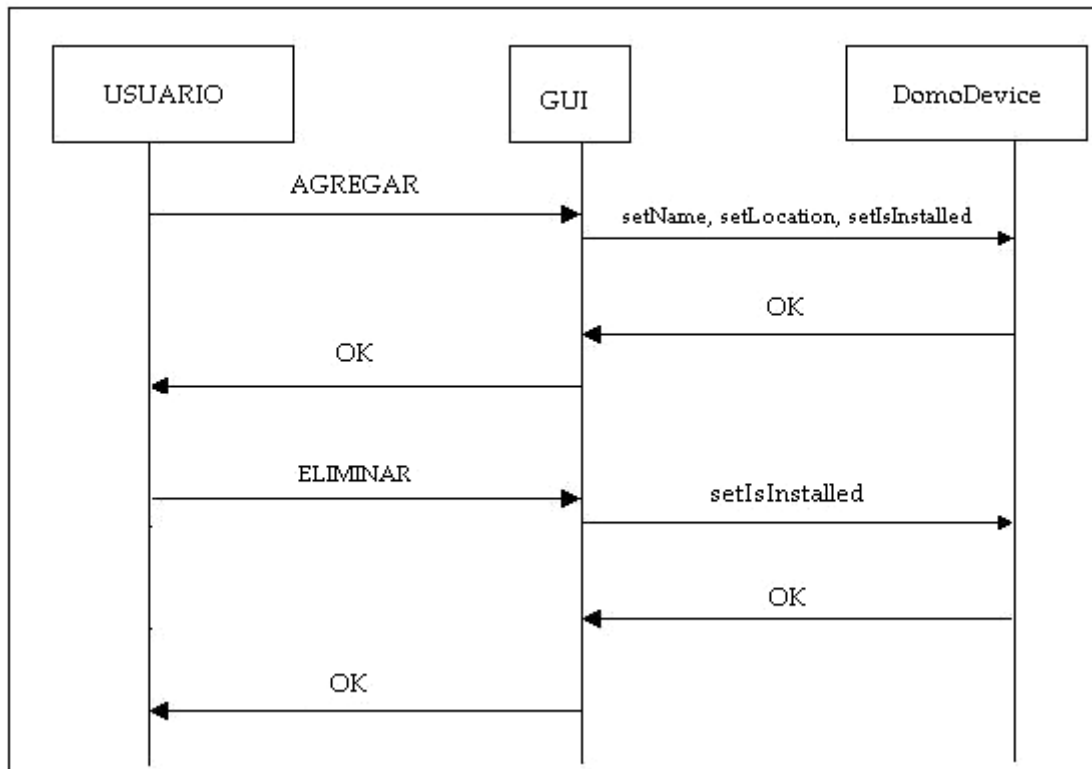
horizontales. Las siguientes imágenes muestran los diagramas de secuencias que surgen en la aplicación:



**Ilustración 25. Diagrama de Secuencias, Gestión Dispositivos Usuario**

La Ilustración 25, detalla la interacción entre los objetos cuando el usuario quiere modificar el estado de cualquier dispositivo domótico. Para ello, siguiendo el ejemplo del dibujo referido a encender una bombilla, el usuario pulsa sobre el botón encender, la GUI envía dos mensajes a un objeto DomoDevice: uno es `setState(value)` para modificar el estado del dispositivo, y otro es `execute()` que sirve para simular la comunicación entre el dispositivo y el bundle del protocolo usado (lonworks, X-10,...) mostrando el nombre y el valor de la acción a ejecutar. Después la GUI

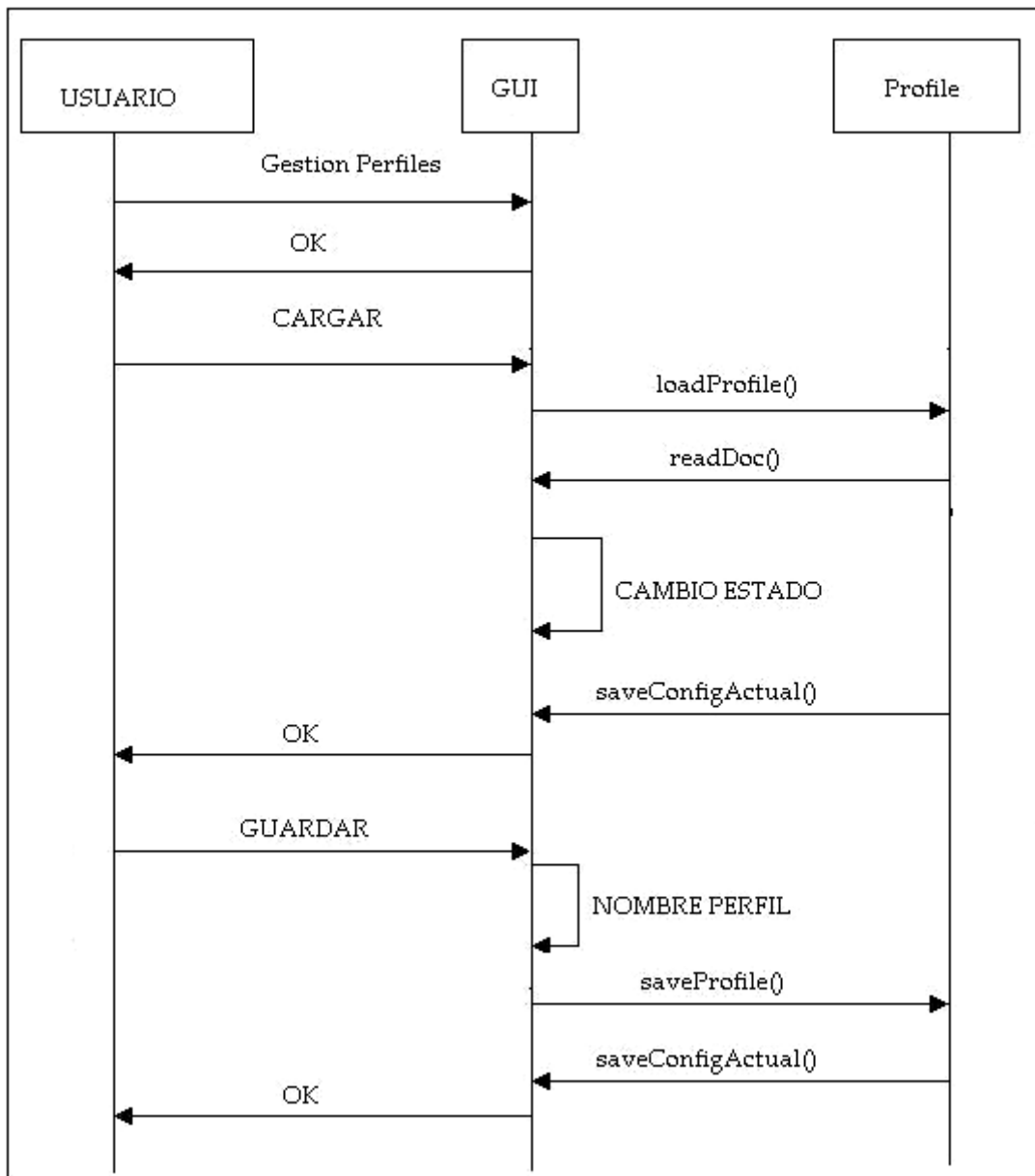
muestra una pantalla informando al usuario del cambio de estado del dispositivo, confirmando la ejecución de la operación.



**Ilustración 26. Diagrama de Secuencias, Gestión Dispositivos Administrador**

La Ilustración 26, detalla la interacción entre los objetos cuando el administrador agrega o elimina un dispositivo de la interfaz. Cuando el administrador pulsa sobre el botón agregar, si ha introducido correctamente los datos de nombre y coordenadas, la GUI envía tres mensajes a un objeto DomoDevice: setName() asigna el nombre dado por el administrador al dispositivo en concreto, setLocation(), asigna a ese dispositivo la localización (y coordenadas) que el administrador le haya asignado, y setIsInstalled(), pone el atributo boolean de dispositivo a true. Cuando el administrador pulsa sobre el botón eliminar, la GUI envía el mensaje

setIsInstalled() a un objeto DomoDevice: cambiando el atributo boolean de dispositivo a false.



**Ilustración 27. Diagrama de Secuencias, Gestión Perfiles**

La Ilustración 27, detalla la interacción entre los objetos cuando el usuario quiere realizar una acción sobre un perfil: guardar, cargar, modificar. Si el usuario quiere cargar o modificar un perfil, el diagrama es el



mismo: pulsa sobre el botón de la acción correspondiente, y la GUI envía un mensaje `loadProfile(name)` a un objeto `Profile`. Este objeto envía un mensaje `readDoc(doc)` al objeto `GUI` que se encarga de actualizar el estado de cada dispositivo. A continuación, la GUI se refresca, mostrando estos cambios, El objeto `Profile` envía un mensaje `saveConfigDeviceActual(file)` al objeto `GUI`, para almacenar la configuración recientemente cargada en el fichero '`ConfigDeviceActual.xml`' (en el siguiente capítulo se explica para qué se usa este fichero). Si el usuario quiere guardar un perfil, al pulsar sobre el botón guardar, la GUI muestra un pantalla para que el usuario introduzca un nombre al perfil y almacena la información de cada dispositivo en un archivo `*.xml`.





## ***6 Implementación***

---

### ***6.1 Servicio DomoDevice***

En OSGi las aplicaciones se construyen en torno a un conjunto de servicios cooperativos. Nuestra aplicación se basa en el registro de un servicio, DomoDevice, por parte de los dispositivos que tenga el hogar, y en la captura de esos servicios por parte de la interfaz gráfica.

El servicio DomoDevice se ha desarrollado según lo estudiado en el capítulo 3 en relación a la clasificación de los dispositivos y sus características. En el cual se estudiaron qué detalles son importantes a la hora de poder interactuar con los dispositivos, como son el ID, estado, tipo, nombre, localización, y saber si está instalado o no. Además se añade la funcionalidad de ejecutar una acción sobre el dispositivo. Pero esta función no está completa ya que se desconoce como es el bundle de un protocolo de control concreto (lonworks, X10,...). Simplemente se implementa un método que debería servir de unión entre el bundle del dispositivo y el del protocolo de control.

```
package org.uc3m.services;

public interface DomoDevice {

    public long getID();

    public Action[] getActions();

    public State getState();

    public void setState(State s);

    public void execute(Action a);

    public String getType();

    public void setName(String name);

    public String getName();

    public void setIsInstalled(boolean b);

    public boolean getIsInstalled();

    public void setLocation(Location location);

    public Location getLocation();

}
```

La imagen anterior muestra el código de la interfaz DomoDevice. En la cual se declaran todas las operaciones (métodos) que podrán realizar los dispositivos.

- Los métodos “getID” y “getType” se usan para obtener el id y el tipo de cada dispositivo.
- Los métodos “setName”, “getName” se usan para gestionar el nombre del dispositivo constituido por un atributo de tipo String.
- Los métodos “setState”, “getState”, se usan para gestionar el estado del dispositivo. El estado de un dispositivo está definido en la clase “State”, constituyendo un estado por dos atributos de tipo String que identifican el valor del estado, y la posición en la que se encuentra. La posición sólo se emplea en dispositivos de persiana, de temperatura, o de luz progresiva (%).
- Los métodos “setLocation”, “getLocation”, se usan para gestionar la localización de un dispositivo. La localización de un dispositivo está definido en la clase “Location”, constituyendo una localización por tres atributos: uno de tipo String , que identifica el espacio donde se encuentra el dispositivo (salón, cocina,...), y dos atributos de tipo int que identifican las coordenadas exactas donde se encuentra ubicado el dispositivo dentro del hogar.
- Los métodos “setIsInstalled”, “getIsInstalled”, se usan para controlar, por un atributo de tipo boolean, si un dispositivo está instalado (agregado a la interfaz).

- El método “getActions”, se usa para obtener las acciones que se ejecutan sobre un dispositivo. Este método debería ser usado sobre el bundle del protocolo empleado en la vivienda (Lonworks, X-10,...) para mandar esa acción al dispositivo y que el dispositivo responda, por ejemplo encendiéndose si la acción es sobre una bombilla y el usuario ha pulsado sobre ‘Encender’.

Las clases de los dispositivos implementarán todos esos métodos, definiendo así el servicio DomoDevice. De los atributos que contienen esas clases, los únicos que se inicializan a un valor por defecto son “ID”, “type”, y “state”. Se establece una configuración por defecto para que al arrancar la aplicación por primera vez los dispositivos ya muestren un estado concreto, todos apagados. El ID y el tipo permanecen constantes durante toda la ejecución de la aplicación.

Como vimos en el apartado 3.5 los *bundles* contienen clases java compiladas. Una de esas clases ejerce de “Activator” del bundle, y contiene un método “start” y un método “stop” para controlar al bundle. Y también vimos que en OSGi las aplicaciones se construyen en torno a un conjunto de servicios cooperativos. Estos están disponibles a través del registro de servicios. En nuestra aplicación el registro del servicio DomoDevice se realiza en las clases “Activator” que se encuentran dentro de los *bundles* de cada dispositivo.

```
DomoDevice dd = new DomoDeviceBlind();  
  
context.registerService("org.uc3m.services.DomoDevice",dd,  
null);
```

Y la captura de este servicio la realiza la clase “Activator” del *bundle* de la interfaz.

```
BundleContext bc=context;ServiceReference sr[];  
  
sr=context.getAllServiceReferences  
  
(DomoDevice.class.getName(),null);
```

## 6.2 *Decisiones de diseño*

Aparte de definir el servicio DomoDevice, se han tenido que tomar decisiones de diseño para realizar la gestión de la aplicación de la forma más eficaz posible. Las principales, son:

### 6.2.1 **Dispositivos instalados en la pasarela residencial, y Dispositivos agregados a la interfaz**

Después de capturar los servicios se lanza la interfaz gráfica. Y a partir de ahí, la información se gestionará principalmente en base a dos arrays de dispositivos: “ddInstalled[]” y “ddTotal[]”. Ambos serán atributos de la clase GUI. El primero de ellos está asociado a los dispositivos que están instalados en la vivienda y reconocidos en la interfaz gráfica (previamente el administrador los debería haber añadido), y el segundo está asociado al total de los dispositivos instalados en la pasarela residencial.

Para obtener el array “ddTotal[]” se recorre el array donde se han capturado el registro de servicios y se obtiene su DomoDevice, almacenándolo en “ddTotal[]”. Este array es usado para saber qué

dispositivos están registrando el servicio en OSGi, y para que el administrador pueda agregarlos a la interfaz si no lo están.

Para obtener el array “ddInstalled[]” se recorre un archivo \*.xml que almacena los dispositivos que el administrador ha agregado anteriormente a la aplicación. En este archivo se almacena su ID, estado, localización, su nombre, y si está instalado. Por tanto, la metodología es identificar cada dispositivo, y almacenar, en los atributos de cada dispositivo, el texto contenido de cada uno de ellos en el archivo, consiguiendo así instalar los dispositivos en su configuración por defecto. Obviamente, los dispositivos añadidos a este array han tenido que, previamente, registrar el servicio DomoDevice, es decir tienen que estar contenidos en el array “ddTotal[]”.

En este punto debemos aclarar que cuando un dispositivo se cae o se instala un nuevo dispositivo (un nuevo bundle) la aplicación se reinicia capturando de nuevo los servicios registrados. Por tanto, debemos diferenciar cuando la aplicación se encuentra en la etapa de arranque y cuando no:

- Si estamos en la etapa de arranque, el array “ddInstalled[]” se obtiene de la forma detallada anteriormente y se lanza la interfaz gráfica mostrando los dispositivos en su configuración por defecto.
- Si no estamos en esta etapa, además de obtener “ddInstalled[]”, hay que devolver a los dispositivos a la configuración existente antes del cambio del número de dispositivos y lanzar la interfaz gráfica.



Para poder devolver los dispositivos a la configuración anterior, siempre que se modifique el estado de algún dispositivo o el usuario cargue un perfil, se almacenará automáticamente la configuración actual de cada dispositivo en un archivo \*.xml (este archivo contendrá el ID y el estado del dispositivo) para, que en caso de caída de un dispositivo o instalación de uno nuevo, acceder a ella. Este fichero será eliminado al salir de la aplicación.

### **6.2.2 Gestión de Perfiles**

Para la gestión de los perfiles de configuración se ha decidido almacenar la información de estado de todos los dispositivos en un archivo \*.xml. Este archivo contiene información acerca del ID y del estado del dispositivo.

Para cargar un perfil se accederá al archivo correspondiente al perfil seleccionado por el usuario, y diferenciando cada dispositivo por su ID se modificará su estado, almacenando en el atributos "state" de cada dispositivo, el texto contenido en el archivo.

También, señalar que la aplicación se ha implementado de tal forma que el usuario puede almacenar, como máximo, 54 perfiles de configuración, pensando que es un número bastante amplio, y no se espera que cause problemas al usuario.

### **6.2.3 Agregar / Eliminar Dispositivos**

Otro punto importante de la aplicación es cuando el administrador accede a ella y añade dispositivos a la interfaz de usuario. Para que pueda

añadir un dispositivo, el administrador, debe seleccionar una ubicación para ese dispositivo. Las posibles ubicaciones dependen de cada casa (si tiene 2 o 3 habitaciones,...), por esa razón se ha creado un archivo 'HomeDefinition' (en formato XML) que define la estructura de la casa. Según lo contenido en ese archivo, el administrador observará unas determinadas ubicaciones donde colocar el dispositivo. De esta forma, se logra que la interfaz sea adaptable a cualquier vivienda.

Además, cuando finalmente el administrador añade o elimina dispositivos a la interfaz, hay que actualizar el "ddInstalled[]". Para ello, se realiza una llamada al método "refreshDDInstalled", que recorre el array "ddTotal[]" y si el método del dispositivo "getIsInstalled" devuelve 'true' lo añade a "ddInstalled[]".



## *7 Pruebas*

---

Las pruebas se integran dentro del ciclo de vida del software y nos permitirá determinar el nivel de calidad que se deben efectuar, además de comprobar el grado de cumplimiento respecto de las especificaciones de requisitos iniciales del sistema. Se dispone de un PC, que hará las veces de pasarela residencial. Dicho PC tiene un procesador Pentium 4 a 3.20 GHz y 512 MB de memoria RAM. En cuanto al software incluye el sistema operativo Windows XP SP1 y la distribución de OSGi Felix.

Para comprobar el buen funcionamiento del programa se ha realizado una batería de pruebas.

- Pruebas de funcionamiento***

Prueba	Resultado	Información Adicional
Gestión Iluminación	ok	El dispositivo cambia de estado y se imprime por pantalla que se ejecuta la acción
Gestión Persianas	ok	El dispositivo cambia de estado y se imprime por pantalla que se ejecuta la acción
Gestión Temperatura	ok	El dispositivo cambia de estado y se imprime por pantalla que se ejecuta la acción
Gestión Fugas de gas	ok	El dispositivo cambia de estado y se imprime por pantalla que se ejecuta la acción
Gestión Acceso	ok	El dispositivo cambia de estado y se imprime por pantalla que se ejecuta la acción
Guardar Perfil	ok	Se crea un archivo *.xml que almacena la configuración actual de los dispositivos
Botón Crear Perfil	ok	La aplicación dirige al usuario al panel Gestión Dispositivos
Modificar Perfil	ok	La aplicación muestra los dispositivos en la configuración almacenada por el correspondiente perfil, permitiendo que el usuario pueda modificarlo.
Cargar Perfil	ok	La aplicación muestra los dispositivos en la configuración almacenada por el correspondiente perfil
Eliminar Perfil	ok	Se elimina el archivo *.xml que indica el perfil seleccionado
Botón Ayuda	ok	La aplicación muestra una ventana emergentes para que el usuario obtenga la ayuda que necesita
Aviso cambio de estado	ok	Cuando el usuario modifica el

dispositivo		estado de un dispositivo, la aplicación muestra una ventana emergente avisando al usuario que ha modificado el estado del dispositivo
Autenticación	ok	El administrador se autentifica en base a dos variables: nombre, y password, almacenados en el archivo 'PasswordAdmin.xml'
Añadir Dispositivo	ok	El administrador añade dispositivos a la aplicación
Eliminar Dispositivo	ok	El administrador elimina dispositivos de la aplicación
Interfaz dinámica	ok	La interfaz se actualiza en vivo, actualizando la lista de dispositivos en función de si están instalados en la plataforma OSGi.

Tabla 43. Pruebas

### Comentarios

El interfaz cumple todos los requisitos funcionales que se recogen en el apartado 4.3.1 de este documento.



## *8 Conclusiones*

---

El entorno de simulación gráfica para servicios OSGi ha quedado concluido. Se ha realizado un estudio de investigación de las tecnologías empleadas en el mundo del hogar digital en base al cual, se ha definido un servicio DomoDevice que permite gestionar la funcionalidad de los principales tipos de dispositivos existentes y las variables que los gestionan. Después se procedió al diseño e implementación de una interfaz gráfica, la cual atiende a unos criterios de usabilidad, que posee todas las

funcionalidades para gestionar el estado de los dispositivos, agregarlos o eliminarlos de la aplicación, y gestionar perfiles de configuración.

Los objetivos marcados antes al comienzo del proyecto, referenciados en el capítulo 2 de este documento, se han conseguido con éxito. El sistema está formado por una interfaz gráfica que permite al usuario final de la aplicación, interaccionar con los dispositivos instalados en su hogar para poder controlar su estado. Además, le permite poder almacenar ciertas configuraciones de los propios dispositivos, para posteriormente poder acceder a ellas, y facilitar la gestión de los dispositivos. Cumpliendo de esta manera con los objetivos 1 y 5.

La elaboración de la interfaz ha sido muy costosa, ya que aparte de diseñar primeramente cada pantalla de acción, hay que tener en cuenta ciertos requisitos de usabilidad, como el tamaño y colocación de los botones, empleando siempre tamaños adecuados y una distribución lógica y sencilla, que facilite la interacción al usuario, cumpliendo con el objetivo 4.

Para desarrollar nuestra aplicación sobre las pasarelas residenciales, cumpliendo con el objetivo 2, convertimos la aplicación en un bundle OSGi, además de exportar sus servicios para ser usados por otro bundles.

Y aprovechando que OSGi ofrece las funciones necesarias para poder añadir y eliminar servicios dinámicamente sin necesidad de resetear el sistema, cumpliendo con el objetivo 3, hemos conseguido que la interfaz sea dinámica, es decir, se actualice en caliente. Ante cualquier variación en los servicios registrados, la aplicación responde mostrando aquellos dispositivos que registran el servicio DomoDevice, evitando así, que la aplicación se vea afectada ante la eventual desaparición de un dispositivo; y



permitiendo la incorporación de nuevos bundles de otros dispositivos a la aplicación.

Además, durante el desarrollo del proyecto hemos avanzado este objetivo, ya que hemos conseguido que la interfaz tampoco dependa de la casa en la que está instalada. Existe un archivo llamado 'HomeDefiniton.xml', que define la estructura de la vivienda, de tal forma que el administrador añadirá los dispositivos en las habitaciones que estén definidas en ese archivo, consiguiendo finalmente una interfaz genérica, adaptable a cualquier situación.

Por último, comentar que la elaboración de este proyecto me ha permitido conocer en profundidad la tecnología OSGi, de la cual no tenía muchos conocimientos, y que es la principal tecnología empleada dentro de éste ámbito de desarrollos. Además, gracias a este trabajo he podido acercarme y profundizar en las tecnologías actualmente empleadas en el mundo del hogar digital y conocer todas las posibilidades que ofrece.



## *9 Líneas futuras de desarrollo*

---

Para ampliar la funcionalidad de nuestra interfaz se pueden realizar varias mejoras, como son:

1. Dar funcionalidad al método “execute” del bundle del dispositivo. Este método es el encargado de comunicarse con el bundle, instalado dentro de OSGi, de Lonworks, X-10, o del protocolo que esté instalado en la vivienda. De esta forma se extiende la

funcionalidad de los bundles de los dispositivos, pudiendo interactuar con ellos físicamente.

2. Ampliación de la aplicación a la gestión de un edificio, diferenciando por plantas y zonas de acceso exclusivo, como pueden ser las viviendas individuales, que se trataran de forma independiente tras un proceso de autenticación de usuario.
3. Que el número de perfiles de configuración almacenados sea ilimitado, porque se ha implementado la posibilidad de almacenar como máximo 54 perfiles (entendiendo que es un número bastante amplio).
4. Definir un nuevo servicio DomoDevice para gestionar dispositivos más complejos como puede ser electrodomésticos, o equipos electrónicos como la televisión, DVD, ...
5. Implementar el uso de sensores, como puede ser un sensor de iluminación que encienda la bombilla en caso de presencia, un sensor de lluvia para que baje las persianas automáticamente, un sensor de gas, para que en caso de fuga cierre la compuerta,...
6. Gestión de diferentes usuarios con diferentes niveles de acceso (sistemas de seguridad, tele-medicina, etc)



## *10 Bibliografía*

---

Para poder realizar este proyecto se necesita una preparación previa, que se obtiene consultando un gran número de fuentes de información. La documentación consultada se muestra en la siguiente lista, numerada según orden de aparición en el texto:

- [1] Domótica [en línea]. <Disponible en:  
<http://es.wikipedia.org/wiki/Dom%C3%B3tica>[Consulta realizada  
el: 06-01-2010]

- 
- [2] Domótica. <Disponible en Web:  
<http://www.domodesk.com/content.aspx?co=51&t=21&c=43> >  
[Consulta realizada el: 08-04-2009]
- [3] Lonworks. <Disponible en Web: <http://odisea.ii.uam.es/>> [Consulta realizada el: 05-04-2009]
- [4] Lonworks. <Disponible en Web: <http://www.casadomo.com/>>  
[Consulta realizada el: 05-04-2009]
- [5] García Izquierdo, José Antonio. Tecnología Lonworks. Junio 2004.  
<Disponible en Web: <http://casafutura.diatel.upm.es/rrssmd/>>  
[Consulta realizada el: 13-01-2010]
- [6] Perfiles funcionales LonMark. <Disponible en Web:  
<http://www.lonmark.org/>> [Consulta realizada el: 19-04-2009]
- [7] Villafuerte, Santiago. Automatización de un hogar mediante tecnología X-10. Escuela Superior de Ingeniería Mecánica y Eléctrica. Unidad Profesional Azcapotzalco. <Disponible en Web: <http://www.migsantiago.com>> [Consulta realizada el: 13-01-2010]
- [8] X-10. <Disponible en Web: <http://domotica-x10.iespana.es/>>  
[Consulta realizada el: 07-04-2009]
- [9] Manual X-10 <Disponible en Web:  
<http://www.aquihayapuntes.com/>> [Consulta realizada el 07-04-2009]
- [10] X10. <Disponible en Web: <http://www.domoticaviva.com/>>  
[Consulta realizada el: 13-01-2010]

- 
- [11] IEEE Medical Device Communications / Health Informatics: Point Of Care Medical Device Communications standard. Part 1. Nomenclature. <Disponible en Web: <http://www.ieee1073.org>.> [Consulta realizada el: 16-07-2009].
- [12] Residential Gateway Architecture and Network Operations.ISO/IEC JTC 1/SC 25/WG 1 N 848, May 1999 <Disponible en Web: <http://hes-standards.org/>> [Consulta realizada el: 16-07-2009].
- [13] Proyecto MUSE. Multi Service Access Everywhere (MUSE) European Project, MUSE-IST 026442< Disponible en Web: <http://ist-muse.org/>> [Consulta realizada el: 16-07-2009].
- [14] Martín Vaquera, Adrián. Sistema de Gestión de citas médicas en entornos de tele-asistencia y tele-seguimiento. Proyecto Fin de Carrera. Universidad Carlos III de Madrid. 2009.
- [15] García Gutiérrez, Bonifacio. Desarrollo de un servicio de acceso y presentación de contenidos multimedia personales e interactivos. Proyecto Fin de Carrera. Universidad Politécnica de Madrid. 2005.
- [16] Fundación Deusto. Programa Saiotek 2006. Entorno de Trabajo Inteligente Colaborativo y Programable. Vigilancia Tecnológica, Plataforma OSGi. <Disponible en Web: <http://www.tecnologico.deusto.es>> [Consulta realizada el:13-01-2010]
- [17] Lorenzo Alcala, Mario. Medida de la usabilidad en aplicaciones de escritorio. Un método práctico. Proyecto Fin de Carrera. Universidad Nacional de Educación a distancia. 2006.

- [18] AEN/CTN 139 Tecnologías de la Información y las Comunicaciones para la Salud. Norma UNE 139801 - 2003 : Aplicaciones informáticas para personas con discapacidad. Requisitos de accesibilidad al ordenador. Hardware. <Disponible en Web: <http://www.udc.es>> [Consulta realizada el:13-01-2010]
  
- [19] AEN/CTN 139 Tecnologías de la Información y las Comunicaciones para la Salud. Norma UNE 139803:2004, “Aplicaciones informáticas para personas con discapacidad. Requisitos de accesibilidad para contenidos en la Web”, a través del Centro de Referencia en Accesibilidad y Estándares Web <Disponible en Web: <http://www.inteco.es/>> [Consulta realizada el:13-01-2010]
  
- [20] Java. <Disponible en Web: <https://www.suntrainingcatalogue.com> [Consulta realizada el 13-01-2010]
  
- [21] Eclipse. <Disponible en Web: <http://www.eclipse.org/org/>> [Consulta realizada el: 24-09-2009]
  
- [22] XML. <Disponible en: <http://www.desarrolloweb.com/articulos/>> [Consulta realizada el: 24-09-2009]
  
- [23] UML. <Disponible en: <http://uml.org/>> [Consulta realizada el: 24-09-2009]
  
- [24] Requisitos Software <Disponible en: <http://www.monografias.com/>> [Consulta realizada el: 01-11-2009]





# *Apéndice I:*

## *Manual de Uso*

---

### *Manual del Administrador*

#### **Instalación de Felix**

Como administrador de nuestra aplicación, usted sabe que funciona sobre el framework de OSGi por ello necesitaremos una plataforma de



arranque, nos servirá varias que existen en el mercado, pero elegiremos Felix por ser una plataforma que cumple con la especificación OSGi R4, además de ser de un proyecto de apache totalmente gratuito.

Puede descargar gratuitamente la plataforma desde la página Web del proyecto apache en la siguiente URL:

<http://felix.apache.org/site/downloads.cgi>

Busque el titulo “Felix Framework Distribución” y en la parte inferior elegiremos la forma de descargarlo dependiendo el sistema operativo Linux “taz.gz” o Windows “.zip”. La plataforma Felix no necesita de ninguna instalación, en el fichero descargado se incluye todas las clases y paquetes necesarios para poder ejecutarlo directamente desde un intérprete de comandos, o como nosotros recomendamos como un proyecto de del la herramienta de desarrollo integrado de código Eclipse, puede usted como usuario seguir fácilmente los pasos indicados en la siguiente URL:


<http://felix.apache.org/site/integrating-felix-with-eclipse.html>

Si usted elige esta opción, simplemente tendrá que ejecutar el proyecto y aparecerá en la consola, la ventana de línea de comandos de Félix. En el caso de que usted elija la opción de abrir un intérprete de comandos tanto en Linux como en Windows, usted tendrá que ir a la ruta donde a descargado el fichero, descomprimirlo y ejecutar la siguiente sentencia:

```
java -jar bin/felix.jar
```

Felix permite trabajar a través de línea de comandos de manera sencilla, su primera impresión nos ofrecerá un mensaje de bienvenida y a

continuación usaremos el comando “help”, para comprobar el resto de comando que usted podrá utilizar. Mostramos una imagen de los que nos debe a aparecer.



```
Símbolo del sistema - java -jar bin/felix.jar
C:\Documents and Settings\usuario\felix-1.0.1>java -jar bin/felix.jar

Welcome to Felix.
=====
->
-> help
bundlelevel <level> <id> ... ! <id> - set or get bundle start level.
cd [<base-URL>] - change or display base URL.
headers [<id> ...] - display bundle header properties.
help - display impl commands.
install <URL> [<URL> ...] - install bundle(s).
obr help - OSGi bundle repository.
packages [<id> ...] - list exported packages.
ps [-l ! -s ! -u] - list installed bundles.
refresh [<id> ...] - refresh packages.
resolve [<id> ...] - attempt to resolve the specified bundles.
services [-ul [-al [<id> ...] - list registered or used services.
shutdown - shutdown framework.
start <id> [<id> <URL> ...] - start bundle(s).
startlevel [<level>] - get or set framework start level.
stop <id> [<id> ...] - stop bundle(s).
uninstall <id> [<id> ...] - uninstall bundle(s).
update <id> [<URL>] - update bundle.
version - display version of framework.
->
```

Ilustración 28. Pantalla de bienvenido a Felix

Algunos de estos comandos, y los que usted va a utilizar son los siguientes:

### **ps**

Muestra una lista de todos los bundles instalados, acompañados de un identificador y su estado actual (Installed, Resolved, Activated)

### **install <ruta>**

Instala el bundle indicado. Para que un bundle pueda instalarse es necesario que se comprima en .jar, con un manifest y un activator.

### **uninstall <id>**

Desinstala el bundle indicado en el campo id

**start <id>**

Arranca el bundle indicado en el campo id

**stop<id>**

Detiene el bundle que posee el identificador <id>

**update<id><ruta>**

Actualizar el bundle <id> con

## Instalación de los bundles

Una vez que usted ha instalado Felix y se familiariza con los comandos más habituales, procederemos a instalar los siguientes *bundles* de la aplicación:

- *DomoDeviceDefinition-1.0.0.jar* : *Bundle* que contiene la implementación del servicio DomoDevice, y otras clases de ayuda como la clase “State”, “Location” y “Action”.
- *DomoDeviceLight\_ID\_150-1.0.0.jar*: *Bundle* que contiene la implementación de un dispositivo de tipo *Ligth*, que registra el servicio DomoDevice.
- *DomoDeviceLightLevel\_ID\_102-1.0.0.jar*: *Bundle* que contiene la implementación de un dispositivo de tipo ‘*Ligth Level*’, que registra el servicio DomoDevice.

- *DomoDeviceBlind\_ID\_124-1.0.0.jar*: *Bundle* que contiene la implementación de un dispositivo de tipo '*Blind*', que registra el servicio *DomoDevice*.
- *DomoDeviceBlind\_ID\_978-1.0.0.jar*: *Bundle* que contiene la implementación de un dispositivo de tipo '*Blind*', que registra el servicio *DomoDevice*.
- *DomoDeviceTemperature\_ID\_023-1.0.0.jar*: *Bundle* que contiene la implementación de un dispositivo de tipo '*Temperature*', que registra el servicio *DomoDevice*.
- *DomoDeviceEntryExit\_ID\_452-1.0.0.jar*: *Bundle* que contiene la implementación de un dispositivo de tipo '*EntryExit*', que registra el servicio *DomoDevice*.
- *DomoDeviceGas\_ID\_785-1.0.0.jar*: *Bundle* que contiene la implementación de un dispositivo de tipo '*Gas*', que registra el servicio *DomoDevice*.
- *GraphicsInterface-1.0.0.jar*: *Bundle* que contiene la implementación de la interfaz gráfica, gestiona los perfiles de configuración, y que recoge todos los dispositivos que registran el servicio *DomoDevice*.

Simplemente usted tiene que escribir el comando *install* y la ruta, se recomienda que usted guarde los *bundle* en la carpeta de "C:\ felix-1.0.1\bundle" así los siguientes comandos le servirán para su sistema de ficheros:

*install file:bundle/DomoDeviceDefinition-1.0.0.jar*

*install file:bundle/DomoDeviceLigth\_ID\_150-1.0.0.jar*

*install file:bundle/DomoDeviceLigthLevel\_ID\_102-1.0.0.jar*

*install file:bundle/DomoDeviceBlind\_ID\_124-1.0.0.jar*

*install file:bundle/DomoDeviceBlind\_ID\_978-1.0.0.jar*

*install file:bundle/DomoDeviceTemperature\_ID\_023-1.0.0.jar*

*install file:bundle/DomoDeviceEntryExit\_ID\_452-1.0.0.jar*

*install file:bundle/DomoDeviceGas\_ID\_785-1.0.0.jar*

*install file:bundle/GraphicsInterface-1.0.0.jar*

Vemos en la siguiente imagen como debe de quedar por el interprete de comando de Felix:



```
C:\Documents and Settings\usuario\felix-1.0.1>java -jar bin/felix.jar

Welcome to Felix.
=====

ps
START LEVEL 1
ID State Level Name
[ 0] [Active] 1 [ 0] System Bundle <1.0.1>
[ 1] [Active] 1 [ 1] Apache Felix Shell Service <1.0.0>
[ 2] [Active] 1 [ 1] Apache Felix Shell TUI <1.0.0>
[ 3] [Active] 1 [ 1] Apache Felix Bundle Repository <1.0.0>
[ 4] [Active] 1 [ 1] DomoDeviceDefinition <1.0.0>
[ 5] [Active] 1 [ 1] DomoDeviceLight_ID_150 <1.0.0>
[ 6] [Active] 1 [ 1] DomoDeviceLightLevel_ID_102 <1.0.0>
[ 7] [Active] 1 [ 1] DomoDeviceBlind_ID_124 <1.0.0>
[ 8] [Active] 1 [ 1] DomoDeviceBlind_ID_978 <1.0.0>
[ 9] [Active] 1 [ 1] DomoDeviceGas_ID_785 <1.0.0>
[10] [Active] 1 [ 1] DomoDeviceEntryExit_ID_452 <1.0.0>
[11] [Active] 1 [ 1] DomoDeviceTemperature_ID_023 <1.0.0>
[12] [Active] 1 [ 1] GraphicsInterface <1.0.0>
```

Ilustración 29. Pantalla de instalación de Bundle

Vemos que los *bundles* están instalados satisfactoriamente por ello procederemos a ejecutarlos. Con el comando start <id> sustituyendo el id por el identificador que nos proporciona Felix para cada *bundle*, ejecutaremos un *bundle*.

Para registrar el servicio DomoDevice, arrancaremos uno por uno los bundles de los dispositivos:

- *"DomoDeviceLigth\_ID\_150-1.0.0.jar"*
- *"DomoDeviceLigthLevel\_ID\_102-1.0.0.jar"*
- *"DomoDeviceBlind\_ID\_124-1.0.0.jar"*
- *"DomoDeviceBlind\_ID\_978-1.0.0.jar"*
- *"DomoDeviceTemperature\_ID\_023-1.0.0.jar"*
- *"DomoDeviceEntryExit\_ID\_452-1.0.0.jar"*
- *"DomoDeviceGas\_ID\_785-1.0.0.jar"*

A continuación, ejecutaremos el bundle de la interfaz gráfica: *"GraphicsInterface-1.0.0.jar"*.

## **Manual de Uso para el administrador**

Este manual muestra las acciones de la interfaz para el Administrador una vez arrancada la aplicación.

La primera pantalla que aparecerá es la 'Pantalla Principal'. Esta pantalla, contiene el botón 'Ayuda', en la parte superior derecha. Pulsando

sobre él, podrá obtener información sobre lo que debe realizar. Además, también se encuentra el botón 'Salir' en la esquina superior izquierda. Es importante que lo tenga en cuenta, porque sólo se encuentra en esta pantalla, y es la única manera de abandonar la aplicación.

Si usted es el administrador del sistema, su tarea, después de arrancar la aplicación, es agregar a la interfaz, aquellos dispositivos que se encuentran instalados en la vivienda. Para ello, en la pantalla principal deberá pulsar sobre el botón 'Gestión Dispositivos'.

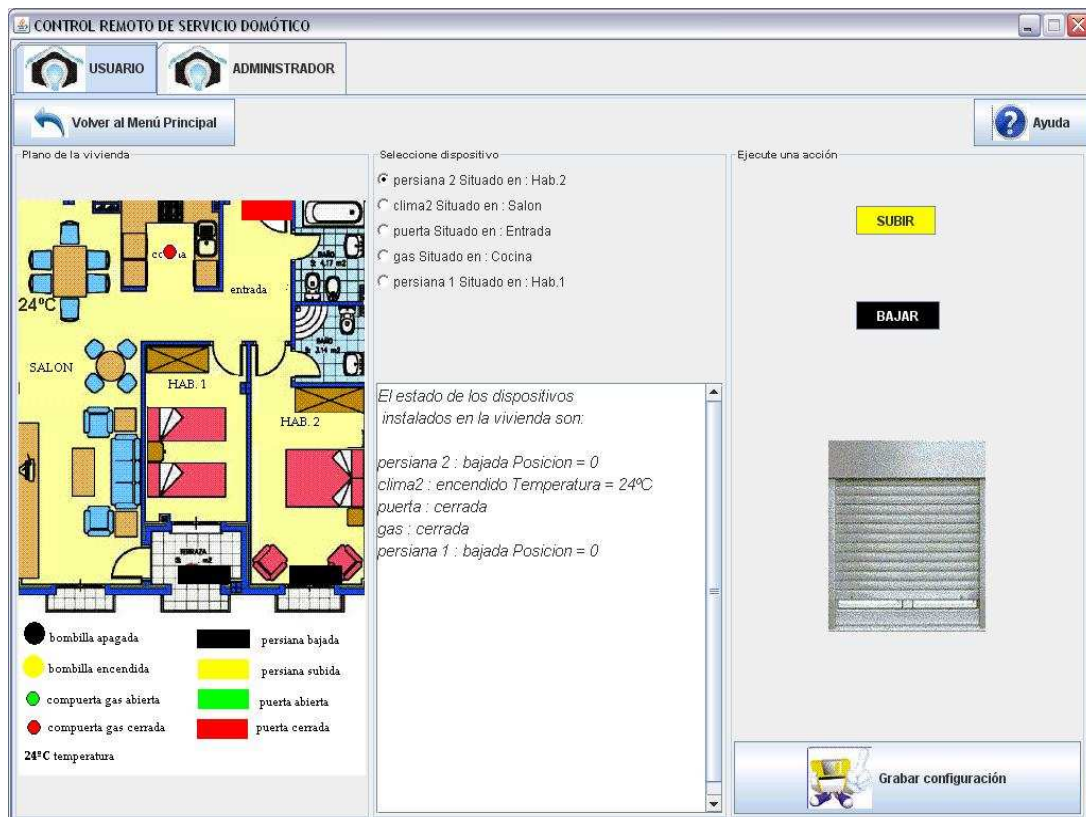


**Ilustración 30. Pantalla principal**

Al pulsarlo, la aplicación le conducirá a la pantalla 'Gestión Dispositivos', y ahí usted, deberá pulsar sobre la pestaña (situada en la parte superior) que contiene el nombre 'ADMINISTRADOR'. Le aparecerá



una ventana pequeña, donde deberá introducir correctamente el Nombre de Usuario y contraseña (admin., admin.) y pulsar sobre el botón 'OK'.



**Ilustración 31. Pantalla 'Gestión Dispositivos'**

A continuación, deberá seleccionar aquél o aquellos dispositivos que necesite agregar, y asignarles una localización y un nombre concreto. Para asignar la localización, como guía se le representa las coordenadas mínima y máxima en el formato (Coordenada X mínima, Coordenada Y mínima – Coordenada X máxima, Coordenada Y máxima) al lado de cada localización, y un mapa de la vivienda en la parte de central de la pantalla. Para agregar el dispositivo, una vez, realizadas todas las tareas anteriores, deberá pulsar sobre el botón 'Agregar Dispositivo' situado en la parte

central (a la derecha se le mostrará los dispositivos que ya se encuentran añadidos, para que le sirva de referencia).

Por último, deberá pulsar sobre el botón 'Guardar', situado en la esquina inferior derecha para que la interfaz reconozca los cambios.

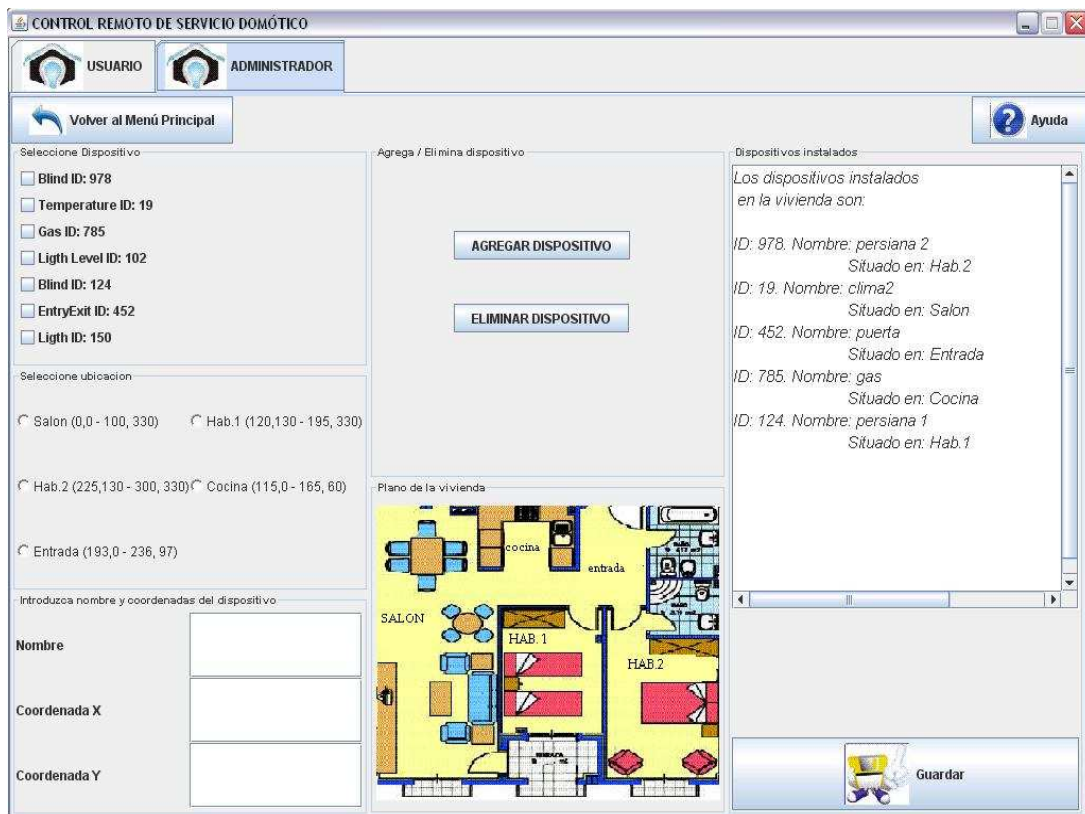


Ilustración 32. Pantalla 'Gestión Dispositivos' - Administrador

## Manual de Usuario

Este manual muestra las acciones de la interfaz para el Usuario una vez arrancada la aplicación.



A lo largo de las 3 pantallas con que cuenta la interfaz, usted tendrá un botón 'Ayuda', en la parte superior derecha. Pulsando sobre él, podrá obtener información sobre lo que debe realizar en cada pantalla.

En la primera pantalla, como muestra la ilustración siguiente, usted puede elegir 2 acciones: 'Gestión Dispositivos' o 'Gestión Perfiles'. La primera le conducirá a la pantalla 'Gestión Dispositivos' con el fin de que usted configure el estado de los dispositivos como prefiera, y pueda almacenar esa configuración en un perfil pulsando sobre el botón 'Guardar configuración' situado en la esquina inferior derecha. La segunda, le conducirá a la pantalla 'Gestión Perfiles' para que usted pueda controlar los perfiles que tiene almacenados en la aplicación, permitiéndole eliminar, cargar, o modificar un perfil existente, o crear uno nuevo. Además, en la pantalla principal, se encuentra el botón 'Salir' en la esquina superior izquierda. Es importante que lo tenga en cuenta, porque sólo se encuentra en esta pantalla, y es la única manera de abandonar la aplicación.



**Ilustración 33. Pantalla principal**

La apariencia de la pantalla 'Gestión Dispositivos' es la que muestra la ilustración X, y como hemos descrito anteriormente permite que usted configure el estado de los dispositivos como prefiera, y pueda almacenar esa configuración en un perfil pulsando sobre el botón 'Guardar configuración'.

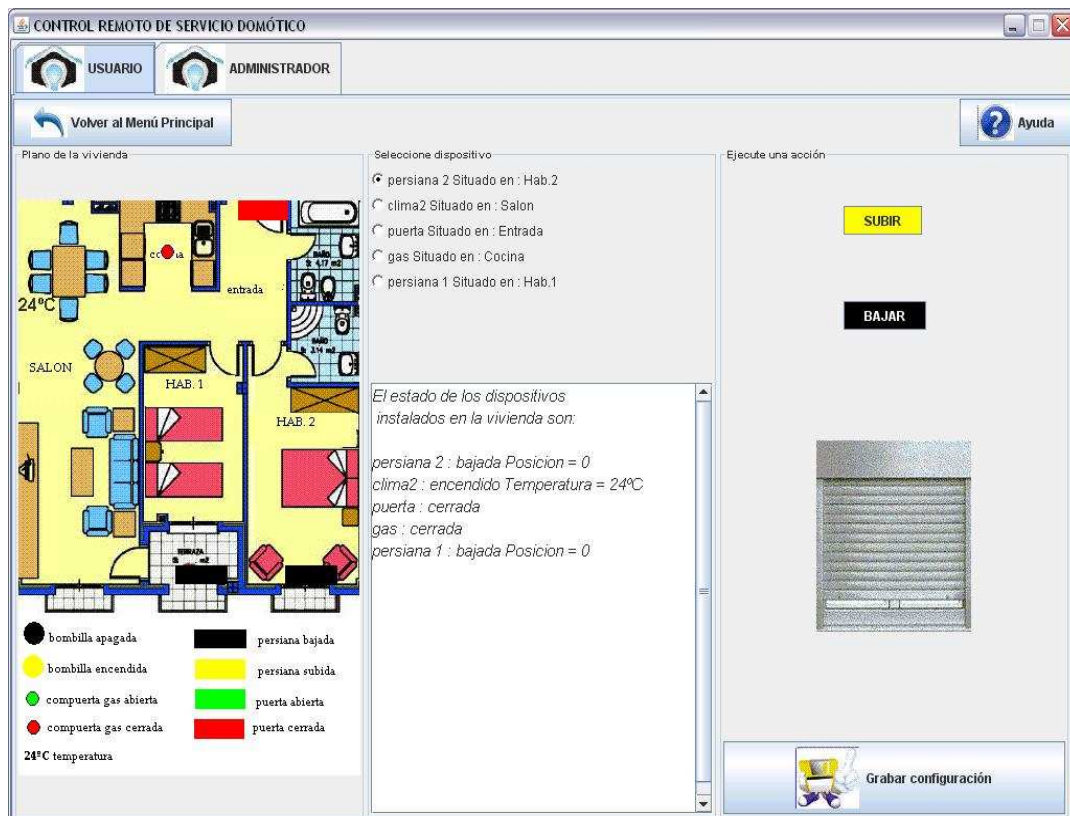
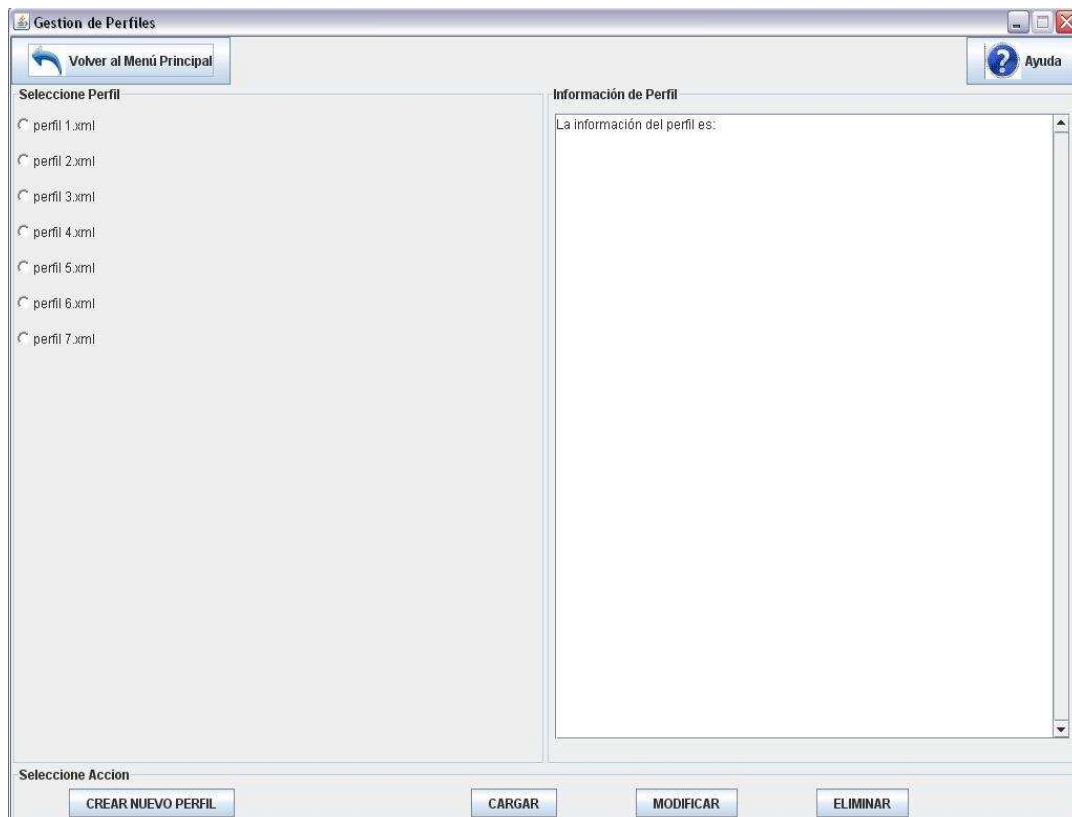


Ilustración 34. Pantalla 'Gestión Dispositivos'

La interacción es sencilla e intuitiva, primero deberá seleccionar el dispositivo al que desea modificar su estado (en la parte inferior se le muestra el estado en que se encuentra cada dispositivo para que le sirva de referencia), y a continuación elige la acción pulsando sobre uno de los botones que aparecen en la parte derecha. Para guardar esa configuración en un perfil, pulsará sobre el botón 'Guardar configuración', y después introducir un nombre a ese perfil. Ante cualquier duda, puede pulsar sobre el botón 'Ayuda' situado en la parte superior derecha, y para volver a la pantalla principal pulsará sobre el botón 'Volver al Menú Principal' situado en la parte superior izquierda.

La apariencia de la pantalla 'Gestión Perfiles' es la que muestra la ilustración X, y como hemos descrito anteriormente permite que usted controle los perfiles que tiene almacenados en la aplicación, pudiendo eliminar, cargar, o modificar un perfil existente, o crear uno nuevo.



**Ilustración 35. Pantalla 'Gestión Perfiles'**

La interacción también es sencilla e intuitiva, primero deberá seleccionar el perfil sobre el que desea realizar una acción. Como referencia, en la parte derecha, se le muestra la información que contiene ese perfil. A continuación, deberá elegir una acción a realizar pulsando sobre el botón que represente dicha acción, que están situados en la parte inferior de la pantalla ('Eliminar', 'Cargar', 'Modificar'). Si su deseo es crear un nuevo perfil, simplemente deberá pulsar sobre el botón 'Crear nuevo Perfil' y la



aplicación le conducirá a la pantalla 'Gestión 'Dispositivos' para que cree una nueva configuración siguiendo la metodología explicada anteriormente. Ante cualquier duda, puede pulsar sobre el botón 'Ayuda' situado en la parte superior derecha, y para volver a la pantalla principal pulsará sobre el botón 'Volver al Menú Principal' situado en la parte superior izquierda.







## *Apendice II:*

# *Descripción del CD*

---

El CD contiene las siguientes carpetas:

- **workspace:** esta carpeta recoge toda la documentación generada por Eclipse. Dentro de ella hay una carpeta para cada *bundle* de la aplicación. Y dentro de estas carpetas hay dos carpetas:

- *out*: contiene un archivo \*.jar que representa el *bundle*., y una carpeta con el nombre del propio *bundle* que contiene los archivos \*.class.
  - *src*: contiene los ficheros de código.
- **Memoria:** Esta carpeta contiene la memoria del Proyecto Fin de Carrera, titulado “Simulación gráfica de servicios osgi en entornos domóticos”.
- **felix:** Es una copia del programa Feliz, la versión felix-1.0.1. Esta carpeta contiene:
- *Carpeta bin*: contiene un fichero \*.jar que almacena las librerías de OSGi.
  - *Carpeta bundle*: aquí se almacenan los bundles de la aplicación generados por eclipse.
  - *Carpeta conf*: contiene el archivo ‘config.properties’, donde se detalla la configuración del Framework OSGi.
  - *Carpeta images*: contiene las imágenes empleadas en la aplicación.
  - *Carpeta profile*: contiene los perfiles de configuración que genere el usuario de la aplicación.
  - Los archivos ‘HomeDefinition.xml’, ‘DeviceInstalled.xml’, ‘PasswordAdmin.xml’, que definen la estructura de la casa donde se instala la



aplicación, la lista de dispositivos instalados por el administrador, y la contraseña de acceso para el administrador, respectivamente.

